



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI *CHAT* PADA
PLATFORM ANDROID DENGAN MEDIA *INPUT*
BERUPA *CANVAS* DAN *SHAREABLE CANVAS*
UNTUK BEKERJA DALAM SATU *CANVAS*
SECARA *ONLINE***

LUFFI ADITYA SANDY
NRP 5113 100 127

Dosen Pembimbing 1
RIZKY JANUAR AKBAR, S.Kom., M.Eng.

Dosen Pembimbing 2
RIDHO RAHMAN HARIADI, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI *CHAT* PADA
PLATFORM ANDROID DENGAN MEDIA *INPUT*
BERUPA *CANVAS* DAN *SHAREABLE CANVAS*
UNTUK BEKERJA DALAM SATU *CANVAS*
SECARA *ONLINE***

LUFFI ADITYA SANDY
NRP 5113 100 127

Dosen Pembimbing 1
RIZKY JANUAR AKBAR, S.Kom., M.Eng.

Dosen Pembimbing 2
RIDHO RAHMAN HARIADI, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESIS - KI141502

CHAT APPLICATION DESIGN ON ANDROID PLATFORM BY USING CANVAS INPUT MEDIA AND SHAREABLE CANVAS TO WORK ON AN ONLINE CANVAS

LUFFI ADITYA SANDY
NRP 5113 100 127

Supervisor 1
RIZKY JANUAR AKBAR, S.Kom., M.Eng.

Supervisor 2
RIDHO RAHMAN HARIADI, S.Kom., M.Sc.

INFORMATICS ENGINEERING DEPARTMENT
Information Technology Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

**RANCANG BANGUN APLIKASI *CHAT* PADA
PLATFORM ANDROID DENGAN MEDIA *INPUT*
BERUPA *CANVAS* DAN *SHARABLE CANVAS* UNTUK
BEKERJA DALAM SATU *CANVAS* SECARA *ONLINE***

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Rumpun Mata Kuliah Algoritma Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

LUFFI ADITYA SANDY

NRP. 5113 100 127

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Rizky Januar Akbar, S.Kom., M.Eng
NIP. 198701032014041001 (Pembimbing 1)
2. Ridho Rahman Hariadi, S.Kom., M.Sc.
NIP. 198702132014041001 (Pembimbing 2)



**SURABAYA
JUNI 2017**

[Halaman ini sengaja dikosongkan]

**RANCANG BANGUN APLIKASI *CHAT* PADA
PLATFORM ANDROID DENGAN MEDIA *INPUT*
BERUPA *CANVAS* DAN *SHAREABLE CANVAS* UNTUK
BEKERJA DALAM SATU *CANVAS* SECARA *ONLINE***

Nama : Luffi Aditya Sandy
NRP : 51131000127
Jurusan : Teknik Informatika
Fakultas Teknologi Informasi ITS
Dosen Pembimbing 1 : Rizky Januar A, S.Kom., M.Eng.
Dosen Pembimbing 2 : Ridho Rahman H, S.Kom., M.Sc.

ABSTRAK

Saat ini sudah banyak tersebar aplikasi dengan layanan chat yang berjalan pada perangkat Android. Semakin berkembangnya teknologi, diperlukan aplikasi yang memberi pengalaman baru kepada pengguna dalam melakukan interaksi dengan pengguna lain. Aplikasi yang sudah ada umumnya menggunakan keyboard sebagai media input dari pengguna untuk mengirim pesan teks dan masih belum memberikan pengalaman yang lebih kepada pengguna dalam melakukan interaksi.

Android menyediakan sebuah library untuk menampilkan media lukis layaknya sebuah kanvas yaitu Android Graphics Library yang pemanfaatannya saat ini hanya sebatas pembuatan aplikasi menggambar seperti di kanvas. Menggunakan library ini, penulis bermaksud memberikan pengalaman baru kepada pengguna dengan mengembangkan sebuah aplikasi yang memiliki fitur mengirim pesan dengan media input berupa kanvas dan membagikan kanvas ke pengguna lain sehingga kedua kanvas pengguna tersinkronisasi

secara realtime dengan dukungan Firebase sebagai back-end services.

Hasil pengujian terhadap aplikasi menunjukkan bahwa aplikasi berhasil memberikan layanan chat dengan media input berupa kanvas dan menghubungkan dua pengguna dalam sebuah kanvas yang tersinkronisasi. Selain itu, hasil kuisisioner menunjukkan bahwa aplikasi memberikan pengalaman baru kepada pengguna dalam berinteraksi.

Kata Kunci: Android Graphics, Chat, Firebase, Keyboard, Library

CHAT APPLICATION DESIGN ON ANDROID PLATFORM BY USING CANVAS INPUT MEDIA AND SHAREABLE CANVAS TO WORK ON AN ONLINE CANVAS

Name : Luffi Aditya Sandy
NRP : 5113100127
Department : Department of Informatics
Faculty of Information Technology ITS
Supervisor I : Rizky Januar A, S.Kom., M.Eng.
Supervisor II : Ridho Rahman H, S.Kom., M.Sc.

ABSTRACT

Nowdays, there are many applications that provide chat feature for Android devices. As technology advances, An application that provides a new experience to the user for interaction is required. Existing applications generally use keyboard as users input medium to chat each other.

Android provides a library to display painting media like a canvas that is Android Graphics Library. Utilization of it is only for the development of drawing applications such as on canvas. Using this library, author intends to provide new user experience by developing an application that has a chatting feature using canvas as input medium and share canvas to another user, so canvas of both user is synchronized in realtime with Firebase support.

Results of application testing show that application is successfully providing a chat service with canvas input medium and connecting two users in a sychronized canvas. Moreover, the results of the questionnaire indicate that the application succeeded in providing new experiences in interacting.

Keywords: Android Graphics, Chat, Firebase, Keyboard, Library

KATA PENGANTAR

Alhamdulillahirobbil'alamin. Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Rancang Bangun Aplikasi Chat pada Platform Android dengan Media Input Berupa Canvas dan Shareable Canvas untuk Bekerja dalam Satu Canvas Secara Online”**.

Pengerjaan tugas akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis, karena dengan ini penulis dapat belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan selama menempuh perkuliahan di Teknik Informatika ITS.

Selesainya tugas akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT yang telah memberikan kesempatan dan anugerah-Nya selama ini.
2. Keluarga Penulis, Ibunda Nurhaidah, Ayahanda Manwar Jenser, dan adinda Dea Khoirunnisa, serta keluarga besar yang tiada henti memberikan dukungan dan doa.
3. Bapak Rizky Januar Akbar S.Kom., M.Eng. selaku dosen pembimbing I tugas akhir yang telah memberikan bimbingan dan arahan selama pengerjaan tugas akhir.
4. Bapak Ridho Rahman H S.Kom., M.Sc. selaku dosen pembimbing II tugas akhir yang telah memberikan bimbingan dan arahan selama pengerjaan tugas akhir.
5. Seluruh administrator dan pengguna laboratorium Algoritma Pemrograman mas Izzudin, mas Khoiry, mbak Nisrina, mbak Shoffi, Yohana, Ridho, Demy, Arvi, Andre, Adi, Ronald, Dimas, Aldi, Rani, Sabila, Mumul, Atul, Ojan, Byan, dan Reinardus yang menemani dan memberikan dukungan selama pengerjaan tugas akhir ini.
6. Teman seperjuangan angkatan Teknik Informatika 2013 yang menemani setiap lika-liku kehidupan kampus bersama.

7. Seluruh pihak yang tidak bisa saya sebutkan satu persatu yang telah memberikan dukungan selama saya menyelesaikan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Aplikasi <i>Chat</i> Serupa	7
2.2 Android	7
2.3 Android Studio	8
2.4 Firebase	9
2.5 Java	12
2.6 Android Graphics Library	12
2.7 Realm	13
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	15
3.1 Analisis.....	15
3.1.1 Analisis Permasalahan.....	15
3.1.2 Analisis Kebutuhan	19
3.1.3 Deskripsi Umum Aplikasi	20
3.1.4 Kasus Penggunaan.....	21
3.2 Perancangan Aplikasi.....	22
3.2.1 Perancangan Arsitektur	22
3.2.2 Perancangan Alur Kasus Penggunaan	23

3.2.2.1	<i>Log in</i>	25
3.2.2.2	Mengirim Pesan (<i>Offline Canvas</i>).....	26
3.2.2.3	Berbagi Kanvas (<i>Shareable Canvas</i>)	29
3.2.2.4	Menambahkan Teman	31
3.2.2.5	Mencari Akun Pengguna Lain.....	32
3.2.2.6	Mengonfirmasi Permintaan Pertemanan	33
3.2.3	Perancangan Alur Pemrosesan Data.....	35
3.2.3.1	Alur Pengolahan Data Sentuhan pada Kanvas	35
3.2.3.2	Alur Penyimpanan Data pada <i>Offline Canvas</i>	37
3.2.3.3	Alur Perpindahan Data pada Fitur <i>Undo</i> dan <i>Redo Offline Canvas</i>	38
3.2.3.4	Alur Penyimpanan Data pada Berbagi Kanvas	39
3.2.3.5	Alur Perpindahan Data <i>Undo</i> dan <i>Redo</i> pada <i>Shareable Canvas</i>	41
3.2.4	Perancangan Basis Data	43
3.2.4.1	Channels	43
3.2.4.2	Friendship.....	44
3.2.4.3	Messages	45
3.2.4.4	Shareable Canvas	46
3.2.4.5	User Detail	47
3.2.5	Perancangan Antarmuka.....	49
3.2.5.1	Halaman <i>Log in</i>	49
3.2.5.2	Halaman Utama.....	50
3.2.5.3	Halaman <i>Chat</i>	52
3.2.5.4	Halaman Kanvas	53
3.2.5.5	Halaman Pencarian.....	54
BAB IV IMPLEMENTASI.....		57
4.1	Lingkungan Implementasi.....	57
4.1.1	Lingkungan Implementasi Perangkat Keras.....	57
4.1.2	Lingkungan Implementasi Perangkat Lunak	57
4.2	Implementasi Alur Kasus Penggunaan.....	58
4.2.1	<i>Log in</i>	58
4.2.2	Mengirim Pesan.....	60
4.2.3	Berbagi Kanvas	63
4.2.4	Mencari Akun Pengguna Lain.....	67

4.2.5	Menambahkan Teman	69
4.2.6	Mengonfirmasi Permintaan Pertemanan	70
4.3	Implementasi Alur Penyimpanan	71
4.3.1	Implementasi Pengolahan Data Sentuhan pada Kanvas	72
4.3.2	Implementasi Penyimpanan Data pada <i>Offline Canvas</i>	74
4.3.3	Implementasi Alur Perpindahan Data pada Fitur <i>Undo</i> dan <i>Redo Offline Canvas</i>	75
4.3.4	Implementasi Alur penyimpanan Data pada <i>Shareable Canvas</i>	76
4.3.5	Implementasi Alur Perpindahan Data pada Fitur <i>Undo</i> dan <i>Redo Shareable Canvas</i>	77
4.4	Implementasi Basis Data.....	79
4.4.1	Implementasi <i>Node Channels</i>	79
4.4.2	Implementasi <i>Node Friendship</i>	80
4.4.3	Implementasi <i>Node Messages</i>	80
4.4.4	Implementasi <i>Node Shareable Canvas</i>	81
4.4.5	Implementasi <i>Node User Detail</i>	81
4.5	Implementasi Antarmuka	82
4.5.1	Halaman <i>Log in</i>	82
4.5.2	Halaman Utama.....	83
4.5.3	Halaman <i>Chat</i>	83
4.5.4	Halaman Kanvas	84
4.5.5	Halaman Pencarian.....	85
BAB V	PENGUJIAN DAN EVALUASI	87
5.1	Lingkungan Pelaksanaan Pengujian.....	87
5.2	Skenario Pengujian	87
5.2.1	Pengujian Fungsionalitas.....	87
5.2.1.1	Pengujian Fitur <i>Log in</i>	88
5.2.1.2	Pengujian Fitur Mengirim Pesan.....	89
5.2.1.3	Pengujian Fitur Berbagi Kanvas	92
5.2.1.4	Pengujian Fitur Mencari Akun Pengguna Lain	94
5.2.1.5	Pengujian Fitur Menambahkan Teman	96

5.2.1.6	Pengujian Fitur Mengonfirmasi Permintaan Pertemanan	98
5.2.2	Pengujian Aplikasi terhadap Pengguna	100
5.3	Evaluasi Pengujian	104
5.3.1	Evaluasi Pengujian Fungsionalitas	104
5.3.2	Evaluasi Pengujian terhadap Pengguna.....	104
BAB VI	KESIMPULAN DAN SARAN.....	107
6.1	Kesimpulan	107
6.2	Saran.....	108
DAFTAR	PUSTAKA.....	109
LAMPIRAN A.	HASIL KUISIIONER	111
BIODATA	PENULIS.....	131

DAFTAR GAMBAR

Gambar 3.1 Ilustrasi <i>Chat</i> dengan Media <i>Input</i> Kanvas	17
Gambar 3.2 Ilustrasi Berbagi Kanvas.....	18
Gambar 3.3 Desain Protokol	19
Gambar 3.4 Diagram Kasus Penggunaan.....	21
Gambar 3.5 Arsitektur Sistem	23
Gambar 3.6 Diagram Alir Penggunaan	24
Gambar 3.7 Diagram Aktivitas <i>Log in</i>	26
Gambar 3.8 Ilustrasi Pengiriman Data <i>Chat</i>	27
Gambar 3.9 Diagram Aktivitas Mengirim Pesan	28
Gambar 3.10 Ilustrasi Pengiriman data <i>Shareable Canvas</i>	29
Gambar 3.11 Diagram Aktivitas Berbagi Kanvas	30
Gambar 3.12 Diagram Aktivitas Menambahkan Teman.....	32
Gambar 3.13 Diagram Aktivitas Mencari Akun Pengguna Lain	33
Gambar 3.14 Diagram Aktivitas Mengonfirmasi Pertemanan	34
Gambar 3.15 Diagram Alir Pengolahan Data Sentuhan.....	36
Gambar 3.16 Diagram Alir Penyimpanan Data pada <i>Offline Canvas</i>	37
Gambar 3.17 Diagram Alir Perpindahan Data pada Fitur <i>Undo Offline Canvas</i>	38
Gambar 3.18 Diagram Alir Perpindahan Data pada Fitur <i>Redo Offline Canvas</i>	39
Gambar 3.19 Diagram Alir Penyimpanan Data pada <i>Shareable Canvas</i>	40
Gambar 3.20 Diagram Alir Perpindahan Data <i>Undo</i> pada <i>Shareable Canvas</i>	41
Gambar 3.21 Diagram Alir Perpindahan Data <i>Redo</i> pada <i>Shareable Canvas</i>	42
Gambar 3.22 Struktur Basis Data Node Channels	43
Gambar 3.23 Struktur Basis Data Node Friendship	44
Gambar 3.24 Struktur Basis Data Node Messages.....	45
Gambar 3.25 Struktur Basis Data Node pada <i>Shareable Canvas</i>	46
Gambar 3.26 Struktur Basis Data Node User Detail.....	48
Gambar 3.27 Halaman <i>Log in</i>	49

Gambar 3.28 Halaman Utama	50
Gambar 3.29 Halaman <i>Chat</i>	52
Gambar 3.30 Halaman Kanvas	53
Gambar 3.31. Halaman Pencarian	55
Gambar 4.1 Implementasi Node Channels	80
Gambar 4.2 Implementasi Node Friendship	80
Gambar 4.3 Implementasi Node Messages	81
Gambar 4.4 Implementasi Node Shareable Canvas	81
Gambar 4.5 Implementasi Node User Detail	82
Gambar 4.6 Implementasi Halaman <i>Log in</i>	82
Gambar 4.7 Implementasi Halaman Utama	83
Gambar 4.8 Implementasi Halaman <i>Chat</i>	84
Gambar 4.9. Implementasi Halaman Kanvas	85
Gambar 4.10. Implementasi Halaman Pencarian	86
Gambar A.7.1 Kuesioner Fitur Mengirim Pesan Pengguna 1 ...	111
Gambar A.7.2 Kuesioner Fitur Berbagi Kanvas Pengguna 1	112
Gambar A.7.3 Kuesioner Fitur Mengirim Pesan Pengguna 2 ...	113
Gambar A.7.4 Kuesioner Fitur Berbagi Kanvas Pengguna 2	114
Gambar A.7.5 Kuesioner Fitur Mengirim Pesan Pengguna 3 ...	115
Gambar A.7.6 Kuesioner Fitur Berbagi Kanvas Pengguna 3	116
Gambar A.7.7 Kuesioner Fitur Mengirim Pesan Pengguna 4 ...	117
Gambar A.7.8 Kuesioner Fitur Berbagi Kanvas Pengguna 4	118
Gambar A.7.9 Kuesioner Fitur Mengirim Pesan Pengguna 5 ...	119
Gambar A.7.10 Kuesioner Fitur Berbagi Kanvas Pengguna 5	120
Gambar A.7.11 Kuesioner Fitur Mengirim Pesan Pengguna 6	121
Gambar A.7.12 Kuesioner Fitur Berbagi Kanvas Pengguna 6	122
Gambar A.7.13 Kuesioner Fitur Mengirim Pesan Pengguna 7	123
Gambar A.7.14 Kuesioner Fitur Berbagi Kanvas Pengguna 7	124
Gambar A.7.15 Kuesioner Fitur Mengirim Pesan Pengguna 8	125

Gambar A.7.16 Kuesioner Fitur Berbagi Kanvas Pengguna 8	126
Gambar A.7.17 Kuesioner Fitur Mengirim Pesan Pengguna 9...	127
Gambar A.7.18 Kuesioner Fitur Berbagi Kanvas Pengguna 9	128
Gambar A.7.19 Kuesioner Fitur Mengirim Pesan Pengguna 10	129
Gambar A.7.20 Kuesioner Fitur Berbagi Kanvas Pengguna10	130

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional.....	20
Tabel 3.2 Keterangan Kode Kasus Penggunaan.....	22
Tabel 3.3 Spesifikasi Kasus Penggunaan <i>Log in</i>	25
Tabel 3.4 Spesifikasi Kasus Penggunaan Mengirim Pesan.....	27
Tabel 3.5 Spesifikasi Kasus Penggunaan Berbagi Kanvas.....	29
Tabel 3.6 Spesifikasi Kasus Penggunaan Menambahkan Teman.....	31
Tabel 3.7 Spesifikasi Kasus Penggunaan Mencari Akun Pengguna Lain.....	32
Tabel 3.8 Spesifikasi Kasus Penggunaan Mengonfirmasi Pertemanan.....	34
Tabel 3.9 Detail Atribut Friend Id Node Friendship.....	45
Tabel 3.10 Detail Atribut Message Id Node Messages.....	46
Tabel 3.11 Detail Atribut Messages Id Node Shareable Canvas.....	47
Tabel 3.12 Detail Atribut User Id Node User Detail.....	48
Tabel 3.13 Spesifikasi Tampilan Halaman <i>Log in</i>	50
Tabel 3.14 Spesifikasi Halaman Utama.....	51
Tabel 3.15 Spesifikasi Halaman <i>Chat</i>	52
Tabel 3.16 Spesifikasi Halaman Kanvas.....	53
Tabel 3.17. Spesifikasi Halaman Pencarian.....	55
Tabel 5.1 Skenario Pengujian Fitur <i>Log in</i> ke-1.....	88
Tabel 5.2 Skenario Pengujian Fitur <i>Log in</i> ke-2.....	89
Tabel 5.3 Skenario Mengirim Pesan Gambar ke-1.....	90
Tabel 5.4 Skenario Mengirim Pesan Gambar ke-2.....	90
Tabel 5.5 Skenario Mengirim Pesan Gambar ke-3.....	91
Tabel 5.6 Skenario Mengirim Pesan Gambar ke-4.....	92
Tabel 5.7 Skenario Pengujian Berbagi Kanvas ke-1.....	93
Tabel 5.8 Skenario Pengujian Berbagi Kanvas ke-2.....	94
Tabel 5.9 Skenario Pengujian Mencari Akun Pengguna ke-1.....	95
Tabel 5.10 Skenario Pengujian Mencari Akun Pengguna ke-2.....	95
Tabel 5.11 Skenario Pengujian Mencari Akun Pengguna ke-3.....	96
Tabel 5.12 Skenario Menambahkan Teman ke-1.....	97
Tabel 5.13 Skenario Menambahkan Teman ke-2.....	97

Tabel 5.14 Skenario Mengonfirmasi Permintaan Pertemanan ke-198

Tabel 5.15 Skenario Mengonfirmasi Permintaan Pertemanan ke-299

Tabel 5.16 Hasil Pengujian Aplikasi terhadap Pengguna pada Fitur Mengirim Pesan.....101

Tabel 5.17 Hasil Pengujian Aplikasi terhadap Pengguna pada Fitur Berbagi Kanvas102

Tabel 5.18 Evaluasi Pengujian terhadap Pengguna pada Fitur Mengirim Pesan.....104

Tabel 5.19 Evaluasi Pengujian terhadap Pengguna pada Fitur Berbagi Kanvas105

DAFTAR KODE SUMBER

Kode Sumber 4.1 Menampilkan Halaman <i>Log in</i>	58
Kode Sumber 4.2 Autentikasi Menggunakan Akun Gmail	59
Kode Sumber 4.3 Menambahkan Detail Pengguna.....	59
Kode Sumber 4.4 Menampilkan Media <i>Input</i> Kanvas	60
Kode Sumber 4.5 Konversi Kanvas ke Gambar dan Proses Unggah	61
Kode Sumber 4.6 Mengirim Pesan Berisi Alamat Gambar.....	62
Kode Sumber 4.7 Mengirim Pesan Teks.....	62
Kode Sumber 4.8 Menerima Pesan dan Menyimpan di Penyimpanan Lokal.....	63
Kode Sumber 4.9 Inisialisasi Saluran <i>Shareable Canvas</i>	64
Kode Sumber 4.10 Menangani Setiap Jenis Sentuhan	65
Kode Sumber 4.11 Mengirim <i>Path Shareable Canvas</i>	66
Kode Sumber 4.12 Mengirim <i>Path Pada Sharable Canvas</i>	66
Kode Sumber 4.13 Kode Mengambil Semua Data Pengguna.....	67
Kode Sumber 4.14 Menyaring Data Pengguna	68
Kode Sumber 4.15 Menampilkan Jendela Konfirmasi Permintaan Pertemanan	69
Kode Sumber 4.16 Mengirim Permintaan Pertemanan	70
Kode Sumber 4.17 Menampilkan Jendela Konfirmasi Permintaan Pertemanan	71
Kode Sumber 4.18 Mengonfirmasi Permintaan Pertemanan	71
Kode Sumber 4.19 Penanganan Sentuhan Layar	72
Kode Sumber 4.20 Menggambar Objek pada Kanvas	73
Kode Sumber 4.21 Penyimpanan Goresan dalam <i>Path</i>	74
Kode Sumber 4.22 Inisialisasi <i>Redo Path</i> dan Fungsi <i>Undo</i>	75
Kode Sumber 4.23 Penggunaan Fungsi <i>Redo</i>	76
Kode Sumber 4.24 Proses Penyimpanan <i>Path</i> pada <i>Shareable Canvas</i>	77
Kode Sumber 4.25 Inisialisasi <i>Redo Path</i> dan Fungsi <i>Undo</i> pada <i>Shareable Canvas</i>	78
Kode Sumber 4.26 Proses Penggunaan Fitur <i>Redo</i> pada <i>Shareable Canvas</i>	79

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dijelaskan hal-hal yang menjadi latar belakang, permasalahan, batasan masalah, tujuan, metodologi, dan sistematika penulisan yang digunakan dalam pembuatan buku tugas akhir ini.

1.1 Latar Belakang

Memasuki era teknologi saat ini, kebutuhan akan fasilitas penyebaran media informasi yang cepat dan mudah semakin meningkat terutama pada aplikasi Android. Salah satu fasilitas media komunikasi sekaligus tempat bertukarnya informasi yang paling populer dan paling banyak digunakan masyarakat adalah *chat*. Pertukaran informasi dalam fasilitas *chat* saat ini tidak hanya berupa data teks, akan tetapi pengguna juga bisa mengirimkan berkas berupa video, gambar, audio, atau jenis berkas lainnya.

Kebanyakan dari aplikasi *chat* yang telah bermunculan menggunakan *keyboard* sebagai media untuk *input* data yang akan dikirimkan kepada lawan *chat*. Pada tugas akhir ini ditawarkan sebuah media lain yang disediakan oleh salah satu *library* yang terdapat dalam pemrograman Java untuk aplikasi Android yaitu *kanvas*. *Kanvas* menangani proses gambar yang didapat dari gerak sentuhan layar pada perangkat telepon pintar. Hasil menggambar di *kanvas* inilah yang saling dikirim antar pengguna sebagai sumber informasi atau data. Dengan adanya aplikasi ini, pengguna bisa merasakan pengalaman baru dalam melakukan percakapan *online* dengan lawan *chat*.

Selain itu, untuk membuat aplikasi menjadi lebih menarik, disediakan juga fasilitas untuk berbagi *kanvas* dengan lawan *chat* sehingga dua pengguna bisa bekerja dalam satu *kanvas* yang terhubung secara *real time*. Hal ini akan membuat aplikasi menjadi lebih interaktif dan menyenangkan.

1.2 Rumusan Masalah

Rumusan masalah yang terdapat pada tugas akhir ini, antara lain adalah:

1. Bagaimana memanfaatkan Android Graphics Library untuk menghasilkan gambar?
2. Bagaimana melakukan pengiriman gambar hasil pemanfaatan Android Graphics Library secara *real time*?
3. Bagaimana melakukan pengiriman *state* pada kanvas antar dua perangkat Android dan melakukan pembaruan kanvas sesuai dengan *state* terkini pada kedua perangkat yang terhubung?

1.3 Batasan Masalah

Batasan masalah yang terdapat pada tugas akhir ini, yaitu sebagai berikut:

1. Aplikasi yang dibuat adalah aplikasi berbasis Android.
2. Bahasa pemrograman yang akan digunakan pada sisi *client* adalah bahasa pemrograman Java.
3. Aplikasi menggunakan Android Graphics Library sebagai penyedia fungsi utama untuk menangani proses menggambar pada perangkat Android.
4. Aplikasi memanfaatkan layanan Firebase sebagai *backend service* aplikasi *mobile*.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk membangun sebuah aplikasi *chat* menggunakan media yang lebih interaktif yaitu kanvas yang disediakan oleh Android Graphics Library. Selain itu tugas akhir ini juga bertujuan untuk membangun

sebuah aplikasi *chat* yang memiliki layanan pengiriman *state* pada dua perangkat Android yang saling berbagi kanvas.

1.5 Manfaat

Tugas akhir ini menghasilkan suatu aplikasi yang bermanfaat untuk memberikan pengalaman baru bagi pengguna dalam berinteraksi satu sama lain. Aplikasi yang berjalan pada perangkat *mobile* berbasis Android ini memungkinkan pengguna melakukan *chat* dengan media *input* kanvas. Aplikasi ini juga bermanfaat untuk memberikan pengalaman berupa suatu media kanvas yang terhubung dan bisa digunakan secara bersamaan oleh dua pengguna pada masing-masing perangkat Android.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan informasi mengenai Android Graphics Library beserta pustaka pendukung lainnya yaitu, bahasa pemrograman Java dan implementasi Firebase sebagai *backend service* yang menyediakan layanan berupa *real time database* non-SQL, *storage*, *cloud messaging*, dan *authentication*. Selain itu, dalam tahap ini juga dicari studi literatur yang relevan untuk pengerjaan tugas akhir ini yang didapatkan dari buku, internet, artikel, dan materi kuliah yang berhubungan dengan metode yang akan digunakan.

2. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis dan perancangan terhadap aplikasi yang dibuat perangkat bergerak dengan platform Android. Perancangan sistem dilakukan dalam beberapa tahap yaitu analisis kebutuhan fungsional yang kemudian digambarkan dalam sebuah diagram kasus penggunaan, perancangan struktur pada basis data

Firestore, perancangan algoritma untuk implementasi kanvas dalam status *offline*, dan algoritma untuk sinkronisasi fitur berbagi kanvas.

3. Implementasi

Pada tahap ini dilakukan implementasi aplikasi berdasarkan perancangan yang dilakukan sebelumnya. Implementasi dilakukan dengan menggunakan bahasa perograman Java, Android Graphics Library sebagai *library* utama yang menyediakan modul untuk implementasi kanvas, IDE Android Studio, dan Firestore sebagai *backend service*.

4. Pengujian dan Evaluasi

Terdapat beberapa metode yang dapat digunakan untuk melakukan pengujian terhadap perangkat lunak. Pada pengerjaan tugas akhir ini, digunakan pengujian dengan metode White-Box dan Black-Box.

Metode White-Box adalah pengujian dengan meneliti kode-kode pada aplikasi dan menganalisis kesalahan yang terdeteksi. Jika terdapat modul yang menghasilkan keluaran yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris kode, variabel, dan parameter yang terlibat perlu dilakukan pengecekan ulang, perbaikan, dan dilakukan penyusunan ulang.

Metode Black-Box adalah pengujian terhadap fungsionalitas dari perangkat lunak. Metode ini disebut juga pengujian partisi yang memungkinkan perekayasa perangkat lunak mendapatkan serangkaian masukan yang sepenuhnya menggunakan semua persyaratan fungsionalitas untuk suatu program.

Metode Grey-Box adalah metode pengujian perangkat lunak yang mengombinasikan antara metode Black-Box dan White-Box. Metode Grey-Box berusaha mengambil kelebihan dari kedua metode tersebut dan mengurangi kekurangan keduanya.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini. Pada tahap ini juga disertakan hasil dari implementasi metode dan algoritma yang telah dibuat.

1.7 Sistematika Penulisan

Penulisan buku tugas akhir bertujuan untuk mendapatkan gambaran pengerjaan tugas akhir dan diharapkan dapat berguna untuk pembaca yang tertarik melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir ini terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Bab ini berisi proses implementasi dari setiap kelas pada semua modul.

Bab V Pengujian dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

Daftar Pustaka

Merupakan daftar referensi yang digunakan dalam tugas akhir ini.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini dan hasil-hasil dari pengujian aplikasi ini.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan pustaka penunjang yang berkaitan dengan pembuatan tugas akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap aplikasi dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Aplikasi *Chat* Serupa

Beberapa aplikasi dengan layanan chat telah tersebar di kalangan pengguna *smartphone*, antara lain Line, Messenger, WhatsApp. Aplikasi tersebut memanfaatkan kanvas hanya untuk memberikan tambahan gambar atau coretan pada foto yang akan dikirim pengguna. Fitur ini serupa dengan fitur yang diberikan pada aplikasi tugas akhir ini yaitu fitur mengirim pesan dengan media *input* berupa kanvas.

Selain itu, juga terdapat sebuah aplikasi yang telah mengimplementasikan fitur berbagi kanvas antar pengguna. Nama aplikasi ini adalah FiveTalk. Aplikasi ini merupakan aplikasi yang sudah sangat berkembang. Namun, aplikasi ini masih terdapat *error* dalam penggunaannya. Selain itu, aplikasi ini memerlukan usaha lebih dari pengguna dalam pendaftaran akun. Pengguna harus memasukkan ID secara manual sejumlah 20 karakter yang digunakan sebagai nilai unik yang berbeda antar pengguna. Pada kanvas yang disediakan juga belum terdapat fitur penambahan bentuk dua dimensi seperti persegi, lingkaran, atau garis.

2.2 Android

Adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar. Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan

tindakan nyata, misalnya menggeser dan mengetuk objek layar[1]. Android juga menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri agar bisa digunakan oleh bermacam piranti bergerak.

Pengguna Android yang terus meningkat menjadikan salah satu alasan penulis memilih platform ini sebagai basis aplikasi. Terdapat batasan pada pengembangan aplikasi yang salah satunya adalah batasan versi sistem operasi Android. Versi minimal dari sistem operasi Android adalah Android 4.4 Kitkat dengan API level 19, sedangkan maksimal adalah versi 7.1 Nougat dengan API level 25.

Android menjadi platform pilihan untuk mengembangkan aplikasi karena Android merupakan sistem operasi yang *open source* sehingga pengembang bisa dengan mudah membuat program memodifikasi, meningkatkan, dan melakukan pemeriksaan pada aplikasi. Selain itu keuntungan yang didapat dari platform Android yang *open source* adalah banyaknya pengembang Android lainnya sehingga referensi sangat mudah untuk didapatkan.

Pengembangan aplikasi Android juga bisa digunakan pada beberapa pilihan IDE seperti Android Studio, Eclipse, atau IntelliJ IDEA. Pada kasus ini penulis menggunakan Android Studio sebagai IDE pengembang aplikasi yang diproduksi langsung oleh Google sebagai pemilik sistem operasi Android.

2.3 Android Studio

Adalah salah satu *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android yang didasari perangkat lunak IntelliJ IDEA. [2]. Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas *programmer* saat membuat aplikasi Android, seperti sistem pembuatan berbasis Gradle yang fleksibel, emulator yang kaya fitur, lingkungan yang menyatu untuk pengembangan bagi

semua perangkat Android, dan *instant run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.

Android Studio menggunakan sistem bangun Gradle yang merupakan pilihan yang baik karena berkembang pesat dan sangat terintegrasi. Gradle memiliki manajemen struktur dengan ketergantungan yang kuat dan hanya memerlukan usaha yang minim untuk melakukan perluasan pada proyek yang dikembangkan. Android Studio juga memiliki fitur *code completion* yang membuat aplikasi pengembangan tidak perlu melakukan usaha lebih dalam hal penulisan sintaksis.

Android studio juga memiliki fitur untuk melakukan desain antarmuka dengan mudah. Ketika pengembang menulis kode pada file XML untuk tampilan aplikasi, maka Android Studio bisa menampilkan hasil kode tanpa perlu menjalankan aplikasi pada emulator atau perangkat Android. Android Studio terbaru juga telah dirilis dengan versi yang lebih stabil dengan performa yang lebih baik. Salah satunya adalah proses *building* yang cukup cepat yaitu sekitar 30 detik. Android Studio juga telah terintegrasi dengan Firebase sebagai *back-end services* yang digunakan pada tugas akhir ini sehingga pengembang tidak perlu melakukan pengaturan manual pada proyek yang dibuat.

2.4 Firebase

Adalah *Backend as a Service* (BaaS) yang saat ini dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah pengembangan aplikasi *mobile*[3]. Dua fitur menarik dari Firebase adalah Firebase Remote Config dan Firebase Real Time Database. Selain itu juga terdapat fitur pendukung untuk aplikasi yang memerlukan *push notification* yaitu Firebase Notification Console.

Remote Config adalah fitur yang memungkinkan pengembang mengganti beberapa konfigurasi aplikasi Android tanpa harus memberikan perubahan aplikasi via Play Store atau

App Store, sedangkan Firebase Real Time Database adalah fitur yang memberikan sebuah non-SQL yang bisa diakses secara *real time* oleh pengguna aplikasi. Firebase bisa menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan sinkronisasi data segera setelah mendapatkan akses internet.

Firebase Database merupakan penyimpanan basis data non-SQL yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain String, Long, dan Boolean. Data pada Firebase Database disimpan sebagai objek JSON *tree*. Tidak seperti basis data SQL, tidak ada tabel dan baris pada basis data non-SQL. Ketika ada penambahan data, data tersebut akan menjadi *node* pada struktur JSON. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa *node* lainnya yang berisi data pula. Proses pengisian suatu data ke Firebase Database dikenal dengan istilah *push*.

Pengembang bisa melakukan *push* terhadap suatu data objek yang telah dibungkus ke dalam suatu kelas ke Firebase Database. Proses *push* ini menghasilkan *node* yang berisi data sesuai dengan atribut dan nilai dari atribut pada objek yang dikirim. Setiap terjadi perubahan data pada Firebase Database, semua pengguna bisa mendapat pemberitahuan melalui *library* yang ditambahkan pada proyek Android Studio. Hal inilah yang membuat basis data pengguna terhubung secara *real time*. Pengembang harus menspesifikasikan *node* apabila ingin mendapat pemberitahuan untuk setiap perubahan pada *node* tersebut.

Tahapan untuk mendapatkan pemberitahuan dari setiap perubahan pada *node* adalah dengan menginisialisasi suatu objek dari kelas `DatabaseReference`. `DatabaseReference` memiliki fungsi `Listener` yang memberikan respon pada program setiap ada perubahan pada Firebase Database. Fungsi ini merupakan fungsi utama yang digunakan pengembang untuk melakukan manajemen pengiriman data pada kanvas sehingga bisa berjalan secara *real time*.

Selain Firebase Database, Firebase menyediakan beberapa layanan lainnya yang juga dimanfaatkan dalam pengembangan aplikasi tugas akhir ini. Layanan tersebut antara lain Firebase Authentication, Storage, dan Cloud Messaging. Pada pengembangan aplikasi tugas akhir ini, Firebase Authentication merupakan layanan yang digunakan untuk menangani pengenalan pengguna yang masuk ke dalam sistem. Firebase Authentication memungkinkan pengembang untuk manajemen proses masuk pengguna ke dalam sistem. Salah satu metode masuk yang bisa digunakan adalah dengan menggunakan akun Gmail. Setiap proses autentikasi yang berhasil dilakukan pengguna, maka pengguna tersebut diberikan ID sebagai nilai unik untuk masing-masing pengguna. ID ini dimanfaatkan dalam aplikasi sebagai *node* utama yang berisi detail masing-masing pengguna.

Layanan lainnya yang digunakan pada pengembangan aplikasi adalah Firebase Storage. Layaknya sebuah penyimpanan awan, Firebase Storage memungkinkan pengembang untuk mengunggah atau mengunduh sebuah berkas. Proses unduh atau unggah pada Firebase Storage terlepas dari kualitas jaringan sehingga proses unggah atau unduh akan berhenti ketika tidak ada koneksi dan dilanjutkan kembali ketika terhubung dengan internet. Pada pengembangan aplikasi tugas akhir ini, Firebase Storage digunakan untuk menyimpan pesan gambar hasil konversi dari kanvas pengirim. Firebase Storage menyediakan folder untuk masing-masing pengguna dengan label berupa ID pengguna sebagai penyimpanan gambar yang dikirim oleh masing-masing pengguna.

Layanan terakhir yang digunakan dalam pengembangan aplikasi adalah Firebase Cloud Messaging. Firebase Cloud Messaging memberikan layanan pengiriman notifikasi antar pengguna. Proses pengiriman notifikasi melalui *rest API* yang disediakan oleh Firebase Cloud Messaging dengan parameter berupa ID aplikasi yang didapatkan dari Firebase dan sebuah

Token penerima notifikasi. Firebase Cloud Messaging memberikan Token untuk masing-masing perangkat bergerak. Token setiap pengguna disimpan dalam Firebase Database sehingga bisa digunakan sebagai parameter dalam suatu pengiriman notifikasi.

Firebase Cloud Messaging juga menyediakan *library* yang ditambahkan pada proyek Android Studio. *Library* ini memungkinkan pengembang untuk menerima notifikasi dan menangani pembuatan Token. Kelas yang digunakan untuk menerima suatu notifikasi adalah `FirebaseMessagingService`. Kelas ini menyediakan sebuah fungsi yaitu `onMessageReceive` yang akan menerima setiap notifikasi yang dikirim dari pengguna lain. Kelas lainnya yang harus diimplementasi adalah `FirebaseInstanceIdService` yang memiliki sebuah fungsi utama yaitu `onTokenRefresh` sehingga pengembang bisa menangani setiap perubahan Token pada perangkat pengguna.

2.5 Java

Adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program Java tersusun dari bagian yang disebut kelas. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan model objek yang lebih sederhana. Java umumnya dikompilasi ke dalam P-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik dan secara khusus didesain untuk implementasi dependensi yang seminimal mungkin[4].

2.6 Android Graphics Library

Framework Android menyediakan sekumpulan fungsi untuk menggambar bentuk dua dimensi[5]. *Library* ini memungkinkan programmer untuk membuat grafik ke dalam kanvas atau memodifikasi tampilan yang sudah ada. Kanvas adalah antarmuka untuk menuju ke *layer* tempat grafis

sebenarnya digambarkan. Melalui kanvas, gambar secara aktual ditunjukkan dalam sebuah bitmap dasar yang ditempatkan dalam suatu jendela.

Pada *library* ini disediakan beberapa kelas yang mendukung pengembangan aplikasi tugas akhir ini. Kelas yang digunakan antara lain Canvas, Paint, dan Path. Kelas Canvas merupakan kelas untuk menyimpan gambar yang dibuat. Untuk menggambar sesuatu, dibutuhkan empat komponen utama yaitu Bitmap untuk menangani piksel, Canvas untuk menangani proses penggambaran, Shape yang mengatur bentuk gambar yang dibuat pada kanvas, dan Paint untuk menangani warna atau atribut dalam menggambar. Kelas Canvas memiliki fungsi `DrawObject` yang digunakan untuk menggambar suatu objek pada tampilan kanvas pengguna.

Selanjutnya adalah kelas Paint yang berfungsi untuk mengatur atribut dari objek yang digambar. Atribut tersebut seperti warna, ketebalan, atau jenis garis pada gambar. Kelas pendukung lainnya adalah Path. Kelas ini mencakup beberapa kontur yang terdiri dari segmen garis lurus dan kurva. Kelas ini yang digunakan untuk menyimpan setiap sentuhan pengguna sehingga bisa digambar ulang. Komposisi dari ketiga kelas ini yang mendukung aplikasi untuk menangani proses menggambar pada kanvas.

2.7 Realm

Adalah sebuah alternatif penyimpanan data pada yang bisa menggantikan SQLite sebagai penyimpanan data bawaan pada Android. Realm melakukan penyimpanan dengan berorientasi pada objek (*native objects*). Tidak perlu melakukan penyalinan objek dari penyimpanan basis data, memodifikasi, dan menyimpannya lagi karena pengembang bekerja dalam objek yang sesungguhnya. Setiap objek tersinkronisasi pada setiap aktivitas yang dilakukan pada objek tersebut dan data dalam objek akan selalu tersimpan pada aplikasi.

Pada pengembangan aplikasi, Realm dipilih menjadi penyimpanan basis data karena data yang diterima dari Firebase Database berupa data JSON *tree* dan Realm menyediakan fungsi yang memungkinkan pengembang untuk mengonversinya menjadi objek. Realm membantu pengembang mengurangi beberapa tahapan yang biasanya dilakukan ketika menerima data dari *server* seperti proses memasukkan data ke basis data lokal dan melakukan *query* ketika ingin membaca data[6].

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini dijelaskan tahap analisis dan perancangan aplikasi yang dibangun, yaitu aplikasi yang memberikan layanan *chat* dengan media masukan berupa kanvas. Dilakukan beberapa analisis pada bab ini, antara lain analisis permasalahan dan analisis kebutuhan. Analisis permasalahan meliputi bahasan mengenai permasalahan yang dihadapi dalam pengerjaan tugas akhir, sedangkan analisis kebutuhan meliputi bahasan mengenai kebutuhan-kebutuhan dalam membangun perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem untuk membangun perangkat lunak. Pendekatan yang digunakan adalah pendekatan berorientasi objek yang dipresentasikan dalam diagram *Unified Modelling Language* (UML).

3.1 Analisis

Tahap analisis dibagi menjadi empat bagian antara lain analisis permasalahan, analisis kebutuhan, deskripsi umum aplikasi, dan kasus penggunaan.

3.1.1 Analisis Permasalahan

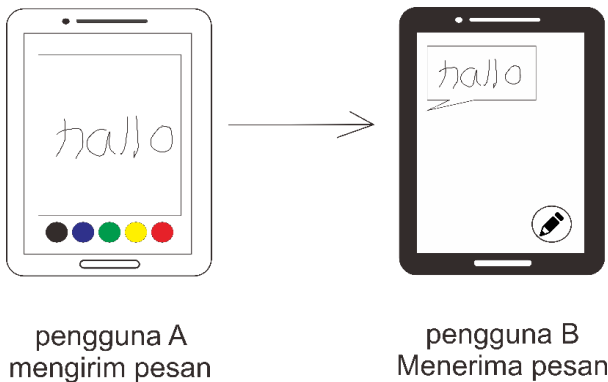
Permasalahan yang diangkat dalam tugas akhir ini adalah memberikan pengalaman baru dalam melakukan *chat* pada aplikasi dengan platform Android. Hal ini disebabkan karena aplikasi *chat* yang telah ada pada umumnya menggunakan *keyboard* sebagai media untuk memasukkan pesan. Untuk membuat aplikasi lebih menarik, aplikasi umumnya juga dilengkapi dengan layanan untuk saling mengirim gambar, pesan, atau video.

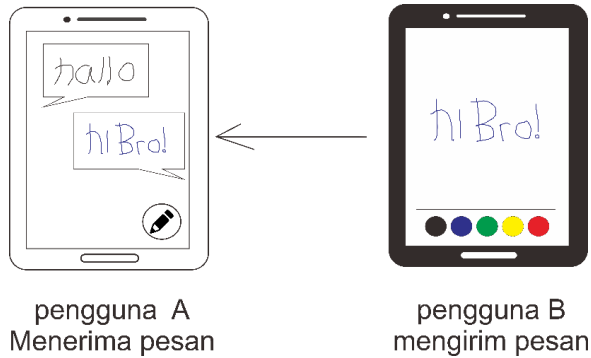
Aplikasi dengan layanan *chat* tentunya memerlukan *backend service* yang berfungsi untuk menyalurkan pesan secara *real time*. Salah satu layanan yang menyediakan *backend* untuk aplikasi *chat* pada Android adalah Firebase. Firebase

menyediakan *real time database* non-SQL yang memungkinkan pengembang mendapatkan pemberitahuan untuk setiap perubahan dalam basis data tersebut.

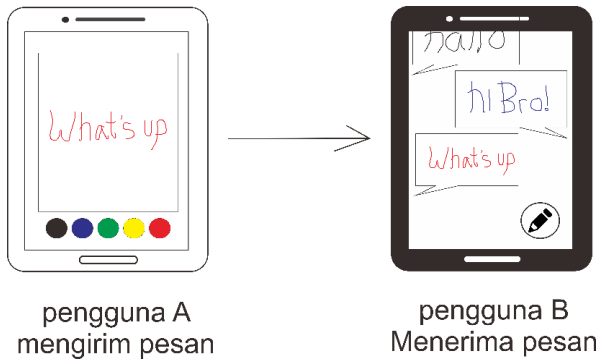
Selain itu, saat ini juga telah tersedia banyak *library* yang bisa digunakan oleh pengembang untuk membangun aplikasi Android, salah satunya adalah Android Graphics Library. *Library* ini menyediakan modul-modul sehingga pengembang bisa mengimplementasikan kanvas dan beberapa modul lainnya pada aplikasi android yang dikembangkan. Namun saat ini, penggunaan Android Graphics Library hanya sebatas aplikasi untuk menggambar pada telepon pintar. Oleh karena itu, untuk memberikan pengalaman baru kepada pengguna smartphone dalam melakukan *chat*, tugas akhir ini memberikan solusi berupa aplikasi yang memanfaatkan kanvas sebagai media input.

Aplikasi yang akan dibuat merupakan aplikasi dengan layanan *chat* yang memiliki fitur utama yaitu media *input* berupa kanvas dan fitur untuk berbagi kanvas bersama lawan *chat* sehingga kedua kanvas pengguna terhubung dan tersinkronisasi secara *real time*. Ilustrasi chat dengan media input berupa kanvas dapat dilihat pada Gambar 3.1.





(2)

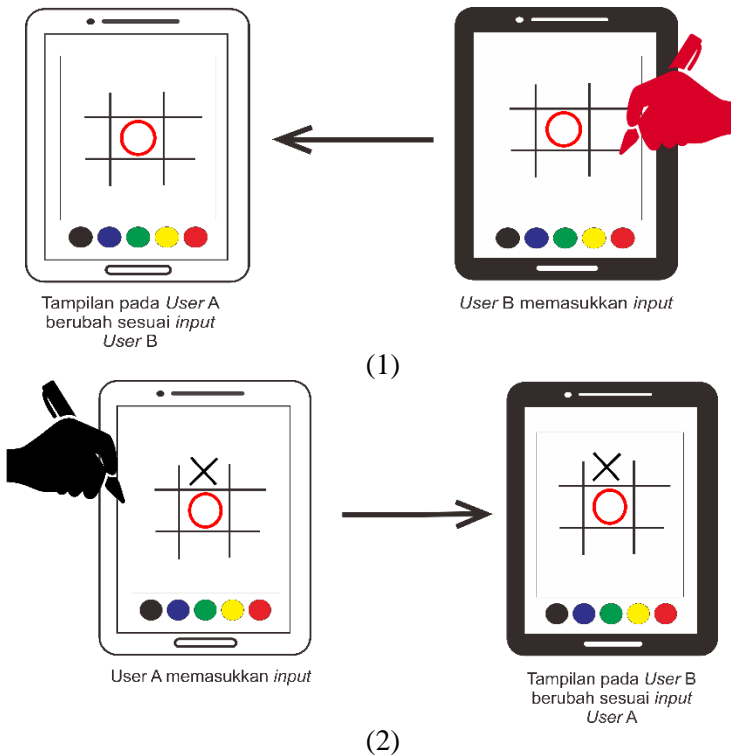


(3)

Gambar 3.1 Ilustrasi Chat dengan Media Input Kanvas

Fitur kedua pada aplikasi adalah berbagi kanvas bersama lawan *chat*. Dua orang pengguna bisa menggunakan satu kanvas yang terhubung. Kanvas di kedua pengguna tersinkronisasi dan diperbarui setiap ada pengguna yang memberikan sentuhan pada kanvas.

Ilustrasi penggunaan fitur berbagi kanvas bisa dilihat pada Gambar 3.2.



Gambar 3.2 Ilustrasi Berbagi Kanvas

Layanan yang akan digunakan untuk melakukan pengiriman pesan adalah Firebase Cloud Messaging. Firebase Cloud Messaging merupakan API yang memungkinkan pengguna untuk mengirim pesan secara aman dari aplikasi yang dikembangkan. Implementasi khusus dari sistem Firebase Cloud Messaging mencakup server aplikasi baik menggunakan HTTP atau protokol XMPP dan aplikasi pengguna. Firebase Cloud Messaging memungkinkan untuk mengirimkan data secara *real time*. Struktur data yang dikirim oleh Firebase Cloud Messaging berupa data JSON.

Dalam aplikasi ini, Firebase menggunakan protokol HTTP dengan format data JSON untuk meneruskan pesan antar aplikasi klien melalui Firebase Cloud Messaging. Desain protokol aplikasi dapat dilihat pada Gambar 3.3.



Gambar 3.3 Desain Protokol

Diharapkan dengan adanya aplikasi ini dapat memberikan pengalaman baru kepada pengguna telepon pintar dalam berhubungan antara satu sama lain.

3.1.2 Analisis Kebutuhan

Kebutuhan utama dari aplikasi ini adalah pengguna dapat mengirimkan pesan berupa gambar dengan media *input* berupa kanvas dan dapat berbagi kanvas dengan lawan *chat* sehingga kedua pengguna bisa bekerja dalam satu kanvas secara *real time*. Kebutuhan fungsional aplikasi tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	<i>Log in</i>	Pengguna bisa masuk ke dalam sistem hanya melalui akun Gmail.
F-0002	Mengirim pesan.	Pengguna dapat mengirim pesan berupa gambar dengan media <i>input</i> berupa kanvas atau berupa teks dengan media <i>input</i> berupa <i>keyboard</i> .
F-0003	Berbagi kanvas	Pengguna dapat berbagi kanvas dengan lawan <i>chat</i> sehingga bisa bekerja dalam satu kanvas secara <i>real time</i>
F-0004	Menambahkan teman	Pengguna bisa menambahkan akun pengguna lain sebagai teman
F-0005	Mencari akun pengguna lain	Pengguna bisa mencari akun pengguna lain dalam daftar pengguna yang telah terdaftar atau pernah menggunakan aplikasi
F-0006	Mengonfirmasi permintaan pertemanan	Pengguna bisa mengonfirmasi untuk menerima atau menolak sebuah permintaan pertemanan.

3.1.3 Deskripsi Umum Aplikasi

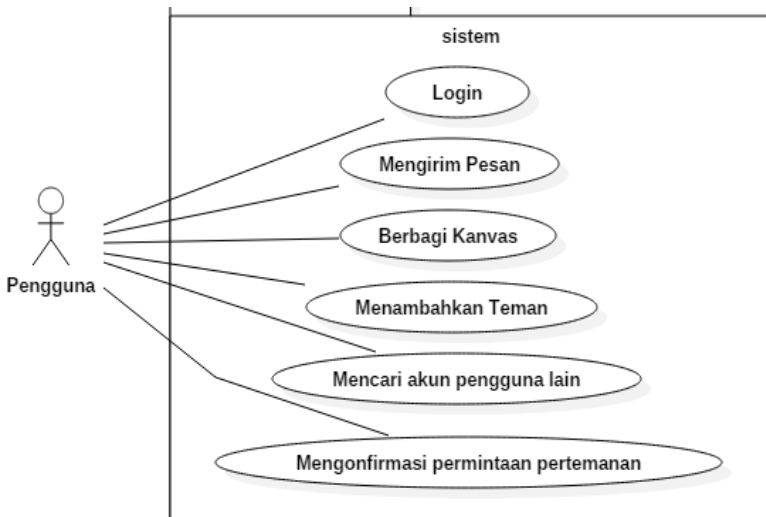
Aplikasi yang dibangun dalam tugas akhir ini adalah sebuah aplikasi pada platform Android dengan layanan utama yaitu *chat*. Yang menjadi pembeda aplikasi ini dengan aplikasi *chat* yang sudah ada yaitu media masukan pesan berupa kanvas.

Pengguna bisa mengirimkan pesan berupa gambar hasil konversi dari sebuah kanvas yang memungkinkan pengguna untuk menggambar di dalamnya.

Selain itu, fitur lain yang juga dapat digunakan adalah saling berbagi kanvas antara dua pengguna. Aplikasi ini menggunakan Android Graphics Library sebagai *library* utama yang menyediakan modul dalam membangun sebuah kanvas. Sedangkan sebagai *backend services*, aplikasi ini didukung penuh oleh Firebase yang memang memiliki layanan-layanan yang dapat membantu dalam pembuatan aplikasi *chat*.

3.1.4 Kasus Penggunaan

Berdasarkan analisis pada kebutuhan fungsional, dibuatlah spesifikasi kasus penggunaan yang diimplementasikan dalam sebuah diagram kasus penggunaan yang dapat dilihat pada Gambar 3.4 dan Tabel 3.2.



Gambar 3.4 Diagram Kasus Penggunaan

Tabel 3.2 Keterangan Kode Kasus Penggunaan

Kode Kasus Penggunaan	Nama
UC-0001	<i>Log in</i>
UC-0002	Mengirim pesan.
UC-0003	Berbagi kanvas.
UC-0004	Menambahkan teman.
UC-0005	Mencari akun pengguna lain.
UC-0006	Mengonfirmasi permintaan pertemanan.

3.2 Perancangan Aplikasi

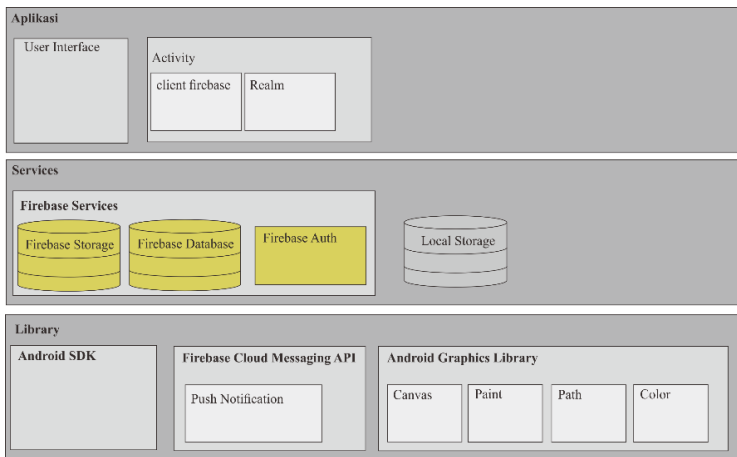
Tahap perancangan dibagi menjadi beberapa bagian antara lain perancangan arsitektur sistem, alur pengiriman data, basis data, alur penggunaan kanvas, dan antarmuka.

3.2.1 Perancangan Arsitektur

Pada arsitektur aplikasi, digambarkan dependensi aplikasi terhadap komponen-komponen di luar aplikasi yang digolongkan ke dalam dua kelompok komponen yaitu *services* dan *library*. *Services* adalah layanan-layanan yang digunakan aplikasi dalam mendukung penyimpanan atau pengiriman data baik lokal maupun antar perangkat, sedangkan *library* adalah kumpulan fungsi untuk memudahkan implementasi program. Arsitektur ini dapat dilihat pada Gambar 3.5.

Beberapa komponen yang tergolong dalam *services* pada proses pembangunan aplikasi yaitu Firebase dan penyimpanan lokal (*local storage*). Firebase menyediakan banyak layanan dalam mendukung pembangunan aplikasi pada perangkat bergerak, namun pada pengembangan aplikasi kali ini, layanan yang digunakan adalah basis data, *authentication*, dan *storage* Firebase. Selain itu juga digunakan penyimpanan lokal pada perangkat bergerak tempat aplikasi dipasang.

Komponen yang tergolong *library* pada pembangunan aplikasi adalah Android SDK sebagai komponen utama pengembangan perangkat Android. Selain itu juga digunakan Firebase Cloud Messaging API yang digunakan untuk mendukung penanganan notifikasi yang dikirim ke perangkat Android, sedangkan Android Graphics Library sebagai penyedia fungsi-fungsi untuk menggambar dalam suatu kanvas pada Android.



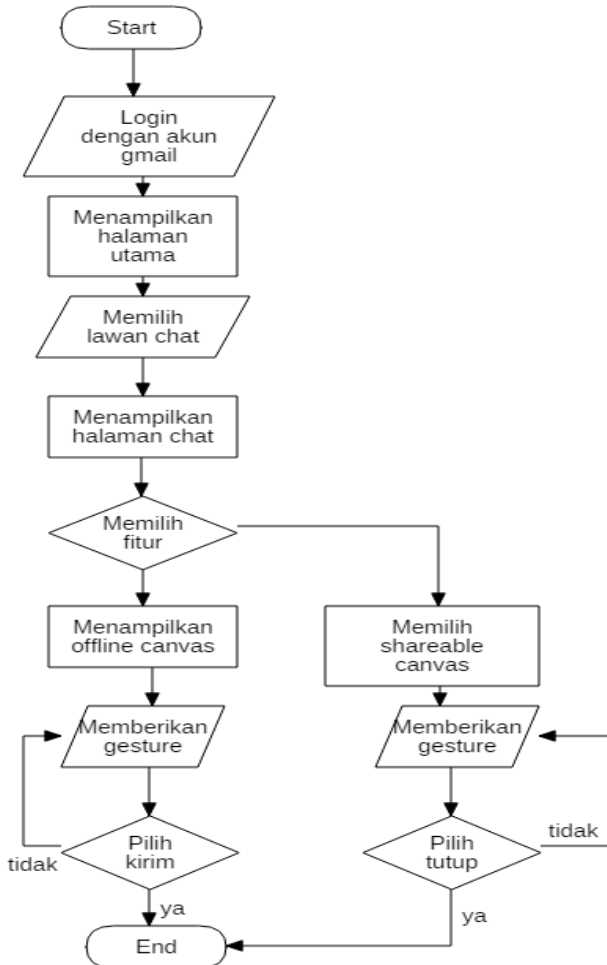
Gambar 3.5 Arsitektur Sistem

3.2.2 Perancangan Alur Kasus Penggunaan

Pada perancangan ini dijelaskan alur penggunaan aplikasi oleh pengguna seperti pada Gambar 3.6 sehingga dapat digunakan sebagai acuan untuk membuat antarmuka aplikasi. Secara umum, pengguna bisa melakukan dua fungsi utama yaitu melakukan *chat* menggunakan kanvas dan membagi kanvas dengan pengguna lain.

Selain dua fitur utama, dijelaskan juga alur untuk setiap kasus penggunaan. Masing-masing alur penggunaan memiliki langkah masing-masing yang harus dilalui oleh pengguna sehingga bisa berjalan sesuai dengan fungsinya. Adapun alur

yang dijelaskan adalah alur kasus penggunaan *log in*, mengirim pesan, berbagi kanvas, menambahkan, dan mengonfirmasi pertemanan.



Gambar 3.6 Diagram Alir Penggunaan

3.2.2.1 Log in

Pada kasus penggunaan ini, pengguna hanya bisa melakukan proses autentikasi menggunakan akun Gmail. Autentikasi diperlukan agar pengguna bisa masuk kedalam sistem untuk menggunakan fitur-fitur yang disediakan.

Jika pengguna belum memiliki akun Gmail, maka pengguna bisa mendaftarkan diri pada layanan yang disediakan oleh Gmail dan telah terhubung langsung dengan sistem sehingga ketika pengguna selesai mendaftar, maka akun tersebut akan terdeteksi oleh sistem untuk menjadi pilihan akun yang bisa digunakan pengguna untuk melalui proses autentikasi.

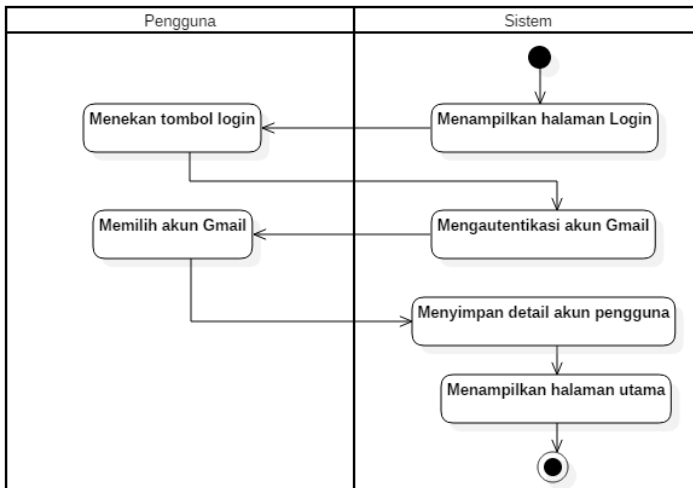
Spesifikasi kasus penggunaan *log in* dapat dilihat pada Tabel 3.3 dan diagram aktivitas pada Gambar 3.7

Tabel 3.3 Spesifikasi Kasus Penggunaan Log in

Kode	UC-001
Nama	<i>Log in</i>
Aktor	Pengguna
Deskripsi	Pengguna melakukan proses autentikasi dengan akun Gmail untuk bisa menggunakan fitur pada aplikasi
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada halaman <i>log in</i>
Kondisi Akhir	Pengguna masuk ke halaman Utama
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Sistem menampilkan halaman <i>log in</i> 2. Pengguna menekan tombol “Log in” 3. Sistem menampilkan jendela baru untuk masuk dengan akun Gmail 4. Pengguna memilih akun Gmail 5. Sistem mengautentikasi akun Gmail 5.5 Pengguna tidak memiliki akun Gmail 6. Sistem menyimpan detil akun pengguna 7. Sistem menampilkan halaman utama 	

Tabel 3.3 Spesifikasi Kasus Penggunaan Log in

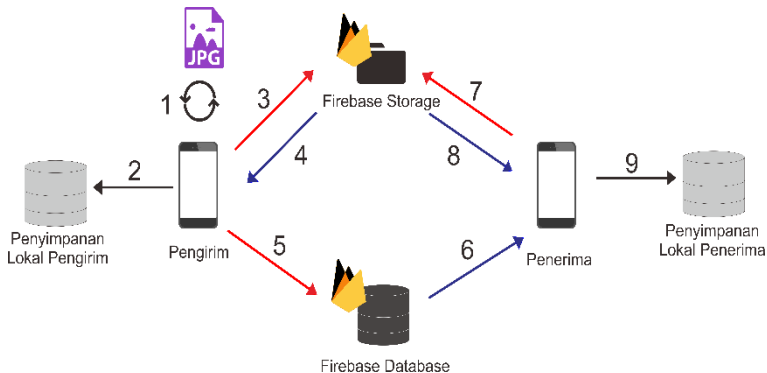
Alur Kejadian Alternatif
A.5. Pengguna tidak memiliki akun Gmail
A.5.1. Pengguna mendaftar sebuah akun Gmail

**Gambar 3.7 Diagram Aktivitas Log in**

3.2.2.2 Mengirim Pesan (*Offline Canvas*)

Pada kasus penggunaan ini, pengguna dapat mengirim pesan berupa teks atau gambar ke pengguna lain dengan beberapa tahapan. Aplikasi menyediakan media *input* untuk membuat pesan yaitu kolom teks dan sebuah *offline canvas*. *Offline canvas* merupakan sebuah media kanvas untuk pengguna membuat gambar sebagai pesan ke pengguna lain. *Offline canvas* disini memiliki arti pengguna tidak membagikan kanvas tersebut ke pengguna lain, hanya sebagai sebuah media *input*. Ilustrasi pengiriman data *chat* dapat dilihat pada Gambar 3.8 dengan melalui beberapa tahapan sebagai berikut:

1. Aplikasi di sisi klien pengirim mengonversi data dalam kanvas menjadi gambar. Gambar hasil konversi disimpan di dalam penyimpanan lokal.
2. Aplikasi mengunggah gambar ke Firebase Storage. Server memberikan respon berupa tautan untuk mengakses gambar.
3. Aplikasi di sisi klien mengirim tautan ke Firebase Database.
4. Firebase Database mengirim data ke penerima.
5. Aplikasi di sisi klien penerima mengakses gambar di Firebase Storage sesuai dengan tautan yang diterima.
6. Aplikasi di sisi klien penerima menyimpan gambar yang di unduh ke dalam penyimpanan lokal.



Gambar 3.8 Ilustrasi Pengiriman Data *Chat*

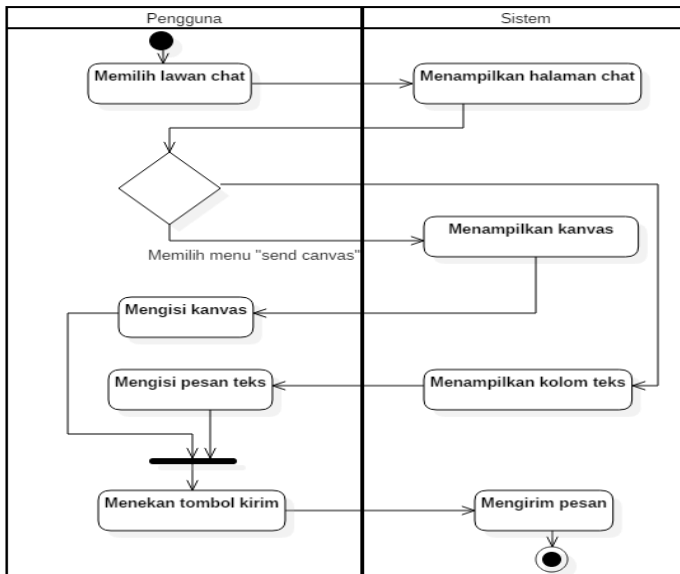
Spesifikasi kasus penggunaan mengirim pesan dapat dilihat pada Tabel 3.4 dan diagram aktivitas pada Gambar 3.9.

Tabel 3.4 Spesifikasi Kasus Penggunaan Mengirim Pesan

Kode	UC-002
Nama	Mengirim pesan
Aktor	Pengguna
Deskripsi	Pengguna mengirim pesan ke lawan <i>chat</i>
Tipe	Fungsional

Tabel 3.4 Spesifikasi Kasus Penggunaan Mengirim Pesan

Kondisi Awal	Pengguna berada di halaman awal
Kondisi Akhir	Pesan terkirim
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna memilih lawan <i>chat</i>. 2. Sistem menampilkan halaman <i>chat</i>. 3. Pengguna memilih menu “Send Canvas”. A.3 Pengguna memilih tombol “Keyboard”. 4. Sistem menampilkan kanvas. 5. Pengguna mengisi kanvas. 6. Pengguna menekan tombol kirim. 7. Sistem mengirim pesan. 	
Alur Kejadian Alternatif	
A.3 Pengguna memilih tombol “Keyboard”. A.3.1 Sistem menampilkan kolom pesan teks. A.3.2 Pengguna mengisi pesan teks.	

**Gambar 3.9 Diagram Aktivitas Mengirim Pesan**

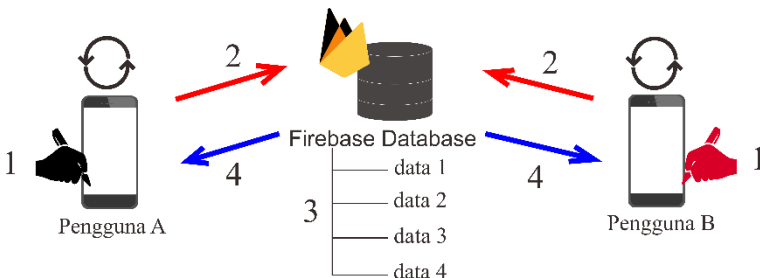
3.2.2.3 Berbagi Kanvas (*Shareable Canvas*)

Pada kasus penggunaan ini dijelaskan mengenai tahapan-tahapan untuk berbagi kanvas dengan pengguna lain. Spesifikasi kasus penggunaan berbagi kanvas dapat dilihat pada Tabel 3.5 dan diagram aktivitas pada Gambar 3.11

Ilustrasi pengiriman data *shareable canvas* dapat dilihat pada Gambar 3.10 dengan melalui beberapa tahapan sebagai berikut:

1. Aplikasi menyimpan data setiap kali pengguna memberikan dan melepaskan sentuhannya pada kanvas.
2. Data tersebut dikirim ke Firebase dan disimpan dalam Firebase Database dalam sebuah antrian data.
3. Antrian diurutkan berdasarkan waktu pengiriman data dari aplikasi ke Firebase.
4. Firebase Database akan meneruskan data ke lawan *chat*.

Setiap perubahan pada Firebase Database akan memicu aplikasi untuk menggambar ulang kanvas sesuai perubahan data.



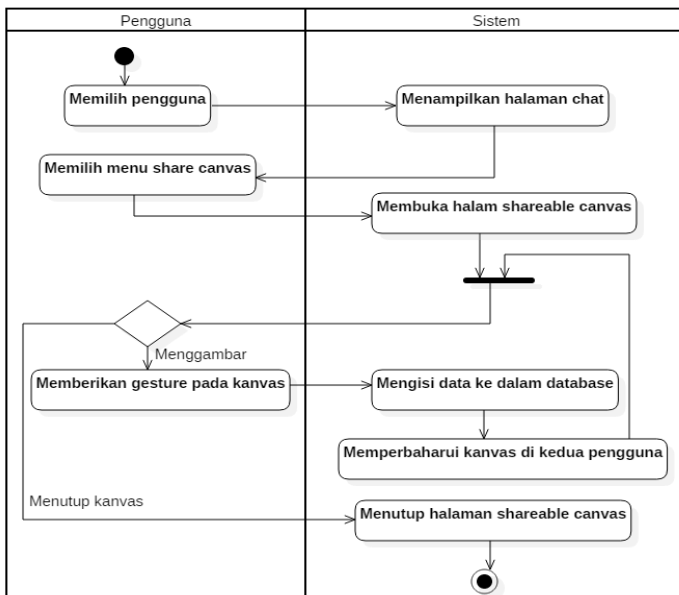
Gambar 3.10 Ilustrasi Pengiriman data *Shareable Canvas*

Tabel 3.5 Spesifikasi Kasus Penggunaan Berbagi Kanvas

Kode	UC-003
Nama	Berbagi Kanvas
Aktor	Pengguna
Deskripsi	Pengguna berbagi kanvas dengan lawan <i>chat</i> .
Tipe	Fungsional

Tabel 3.5 Spesifikasi Kasus Penggunaan Berbagi Kanvas

Kondisi Awal	Pengguna berada di halaman utama.
Kondisi Akhir	Kedua pengguna bekerja dalam satu kanvas secara <i>online</i> .
Alur Kejadian Normal	
<ol style="list-style-type: none"> 1. Pengguna memilih pengguna lain sebagai lawan <i>chat</i>. 2. Sistem menampilkan halaman chat. 3. Pengguna memilih menu “Share Canvas”. 4. Sistem membuka halaman <i>share canvas</i>. 5. Pengguna memberikan <i>gesture</i> pada kanvas. A.5 Pengguna memilih menu tutup. 6. Sistem mengisi data ke dalam basis data. 7. Sistem memperbaharui kanvas di kedua pengguna. 	
Alur Kejadian Alternatif	
A.5. Pengguna memilih menu tutup.	
A.5.1. Sistem menutup halaman <i>shareable canvas</i> .	

**Gambar 3.11 Diagram Aktivitas Berbagi Kanvas**

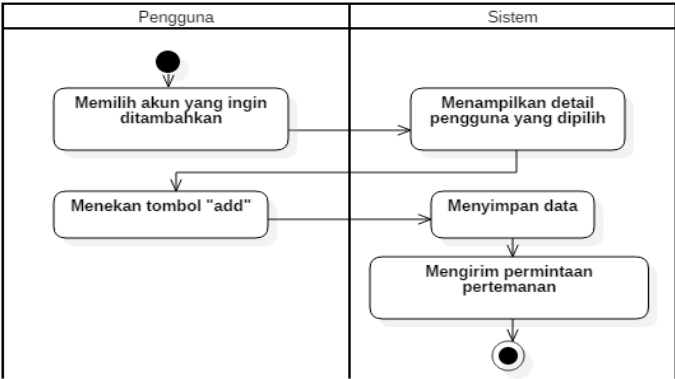
3.2.2.4 Menambahkan Teman

Pada kasus penggunaan ini dijelaskan mengenai tahapan-tahapan untuk menambahkan teman. Pengguna bisa menambahkan teman dari setiap pengguna yang pernah melakukan proses autentikasi pada sistem. Setelah mengirim permintaan pertemanan, pengguna lain menerima permintaan pertemanan dan bisa melakukan konfirmasi dengan menambahkan atau mengabaikan permintaan yang dikirim.

Spesifikasi kasus penggunaan mencari akun pengguna lain dapat dilihat pada Tabel 3.6 dan diagram aktivitas pada Gambar 3.12.

**Tabel 3.6 Spesifikasi Kasus Penggunaan
Menambahkan Teman**

Kode	UC-005
Nama	Menambahkan teman.
Aktor	Pengguna
Deskripsi	Pengguna menambahkan akun pengguna lain sebagai teman.
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada halaman cari.
Kondisi Akhir	Permintaan pertemanan dikirim.
Alur Kejadian Normal	
1. Pengguna memilih akun pengguna lain yang ingin ditambahkan. 2. Sistem menampilkan detail pengguna yang dipilih. 3. Pengguna menekan tombol tambah teman. 4. Sistem menyimpan data. 5. Sistem mengirim permintaan pertemanan.	
Alur Kejadian Alternatif	
Tidak ada	



Gambar 3.12 Diagram Aktivitas Menambahkan Teman

3.2.2.5 Mencari Akun Pengguna Lain

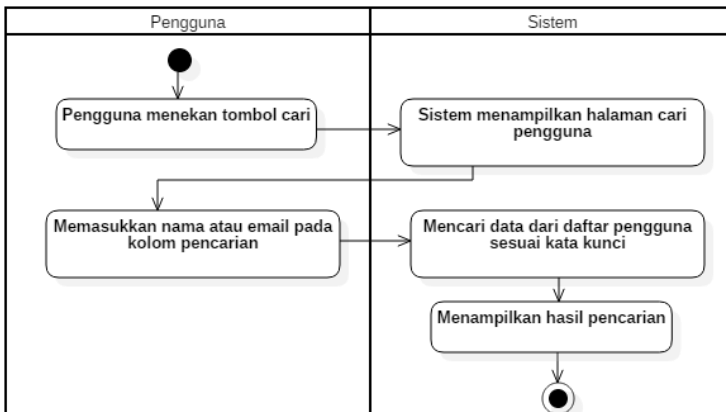
Pada kasus penggunaan ini dijelaskan mengenai tahapan-tahapan untuk mencari akun pengguna lain. Spesifikasi kasus penggunaan mencari akun pengguna lain dapat dilihat pada Tabel 3.7 dan diagram aktivitas pada Gambar 3.13.

Tabel 3.7 Spesifikasi Kasus Penggunaan Mencari Akun Pengguna Lain

Kode	UC-004
Nama	Mencari Akun Pengguna Lain.
Aktor	Pengguna
Deskripsi	Pengguna mencari akun pengguna lain.
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada halaman awal.
Kondisi Akhir	Sistem menampilkan daftar pengguna yang dicari.
Alur Kejadian Normal	
1. Pengguna Menekan tombol cari.	
2. Sistem menampilkan halaman cari pengguna.	
3. Pengguna memasukkan nama atau <i>email</i> pada .	

Tabel 3.7 Spesifikasi Kasus Penggunaan Mencari Akun Pengguna Lain

4. Sistem mencari akun pengguna sesuai kata kunci
5. Sistem menampilkan hasil pencarian
Alur Kejadian Alternatif
Tidak ada

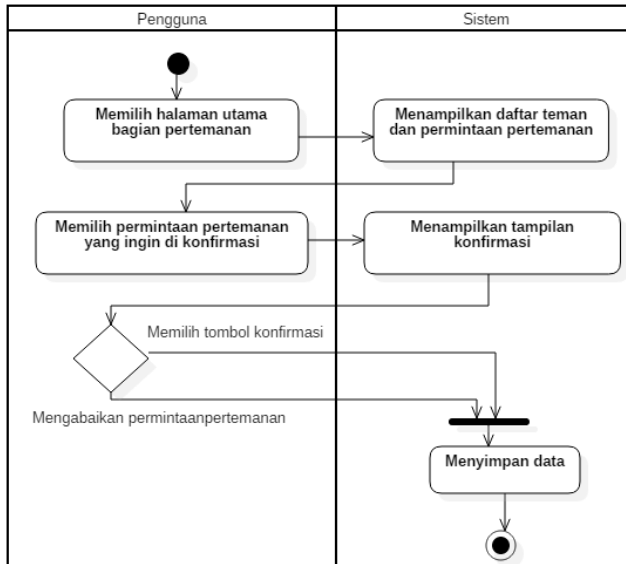


Gambar 3.13 Diagram Aktivitas Mencari Akun Pengguna Lain

3.2.2.6 Mengonfirmasi Permintaan Pertemanan

Pada kasus penggunaan ini dijelaskan mengenai tahapan-tahapan untuk mengonfirmasi permintaan pertemanan. Konfirmasi pertemanan hanya bisa dilakukan pengguna menerima permintaan pertemanan dari pengguna lain. Permintaan pertemanan ditampilkan pada halaman utama aplikasi sehingga pengguna bisa melihat secara langsung apabila terdapat permintaan pertemanan yang baru dikirim.

Spesifikasi kasus penggunaan mengonfirmasi permintaan pertemanan dapat dilihat pada Tabel 3.8 dan diagram aktivitas pada Gambar 3.14.



Gambar 3.14 Diagram Aktivitas Mengonfirmasi Pertemanan

**Tabel 3.8 Spesifikasi Kasus Penggunaan
Mengonfirmasi Pertemanan**

Kode	UC-006
Nama	Mengonfirmasi Pertemanan.
Aktor	Pengguna
Deskripsi	Pengguna mengonfirmasi permintaan pertemanan yang diterima.
Tipe	Fungsional
Kondisi Awal	Pengguna berada pada halaman utama.
Kondisi Akhir	Data konfirmasi tersimpan.
Alur Kejadian Normal	
1. Pengguna memilih halaman pertemanan. 2. Sistem menampilkan daftar teman dan permintaan pertemanan. 3. Pengguna memilih permintaan pertemanan yang ingin di konfirmasi.	

**Tabel 3.8 Spesifikasi Kasus Penggunaan
Mengonfirmasi Pertemanan**

4. Sistem menampilkan jendela untuk mengonfirmasi.
5. Pengguna memilih tombol konfirmasi.
A.5. Pengguna mengabaikan konfirmasi.
6. Sistem menyimpan data.
Alur Kejadian Alternatif
A.5. Pengguna mengabaikan mengonfirmasi.

3.2.3 Perancangan Alur Pemrosesan Data

Perancangan alur pemrosesan data menjelaskan cara sistem dalam memproses data sehingga bisa ditampilkan ke pengguna sesuai fungsinya.

Pada alur pemrosesan data dibagi menjadi beberapa bagian yaitu pengolahan, penyimpanan, perpindahan, dan pengiriman data pada sistem.

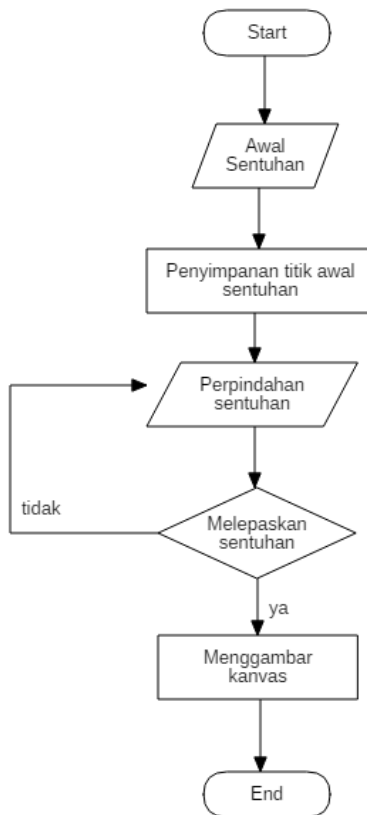
3.2.3.1 Alur Pengolahan Data Sentuhan pada Kanvas

Alur pengolahan data kanvas berisi langkah-langkah aplikasi dalam menerima sentuhan dan menggambarkan sentuhan tersebut ke dalam kanvas yang ditampilkan ke pengguna.

Alur pengolahan data kanvas dapat dilihat pada Gambar 3.15 dengan melalui beberapa langkah sebagai berikut:

1. Aplikasi menyediakan suatu fungsi yang menangani setiap sentuhan yang diterima kanvas. Sentuhan dibagi menjadi tiga jenis yaitu sentuhan awal saat pengguna pertama kali memberi sentuhan, melakukan perpindahan sentuhan, dan saat pengguna melepas sentuhan pada kanvas.
2. Aplikasi menyediakan suatu variabel yang disebut *path* untuk menyimpan satu grafis mulai dari pengguna menyentuh hingga melepas sentuhan.

3. Ketika satu jenis sentuhan diterima, aplikasi melakukan pengecekan jenis gambar yang dipilih pengguna. Jenis gambar dibagi menjadi beberapa bentuk yaitu persegi, lingkaran, garis, dan abstrak (*freehand*).
4. Saat pengguna melepas sentuhan dari kanvas, *path* menggambar bentuk yang dipilih pengguna ke dalam kanvas dengan titik-titik koordinat yang direkam selama sentuhan berlangsung.



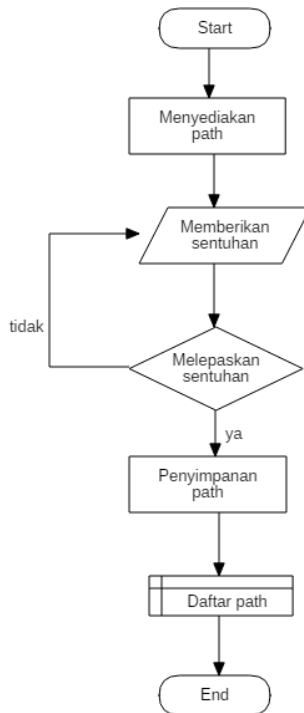
Gambar 3.15 Diagram Alir Pengolahan Data Sentuhan

3.2.3.2 Alur Penyimpanan Data pada *Offline Canvas*

Alur penyimpanan data pada *offline canvas* dapat dilihat pada Gambar 3.16 dengan melalui beberapa tahapan sebagai berikut:

1. Aplikasi menyediakan suatu penyimpanan yang berisi daftar *path* yang disebut *canvas path*.
2. Setiap pengguna melepaskan sentuhan, maka *canvas path* menyimpan *path* tersebut pada urutan terakhir.

Langkah pertama sampai kedua diulang setiap pengguna mulai memberikan *gesture* atau menyentuh layar pada kanvas lagi.



Gambar 3.16 Diagram Alir Penyimpanan Data pada *Offline Canvas*

3.2.3.3 Alur Perpindahan Data pada Fitur *Undo* dan *Redo Offline Canvas*

Alur penyimpanan data pada fitur *undo* dan *redo offline canvas* dapat dilihat pada Gambar 3.17 dan Gambar 3.18 dengan melalui beberapa tahapan. Tahapan-tahapan pada satu kali penggunaan fitur *undo* adalah sebagai berikut:

1. Membuat sebuah variabel yang berisi daftar *path* yang disebut *redo path* untuk menyimpan *path* yang dibatalkan oleh pengguna.
2. Melakukan pengecekan pada *canvas path*. Jika kosong, maka fitur *undo* tidak bisa digunakan.
3. Jika *canvas path* tidak kosong, maka *path* terakhir pada *canvas path* akan dipindahkan ke dalam *redo path*.

Setiap pengguna memberikan sebuah *gesture*, maka semua *path* pada *redo path* akan dihapus.

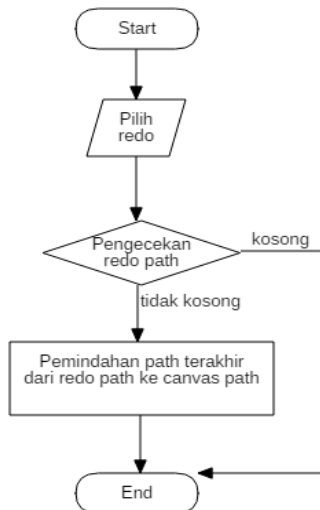


Gambar 3.17 Diagram Alir Perpindahan Data pada Fitur *Undo Offline Canvas*

Tahapan-tahapan pada satu kali penggunaan fitur *redo* adalah sebagai berikut :

1. Melakukan pengecekan pada *redo path*. Jika kosong, maka fitur tidak bisa digunakan.
2. Jika *redo path* tidak kosong, maka *path* terakhir pada *redo path* akan dipindahkan ke urutan terakhir dalam *canvas path*.

Kemudian *canvas* akan menggambar ulang setiap *path* yang ada dalam *canvas path* sesuai dengan urutan masuk.



Gambar 3.18 Diagram Alir Perpindahan Data pada Fitur *Redo Offline Canvas*

3.2.3.4 Alur Penyimpanan Data pada Berbagai Kanvas

Alur penyimpanan data pada berbagi kanvas dapat dilihat pada Gambar 3.19 dengan melalui beberapa tahapan sebagai berikut:

1. Aplikasi menyediakan sebuah *global path* yang menyimpan daftar dari semua *path* yang dibuat oleh kedua pengguna dan sebuah *canvas path* untuk menyimpan semua *path* milik masing-masing pengguna.
2. Ketika pengguna melepaskan sentuhan dan disimpan kedalam sebuah *path*, maka *path* tersebut akan disimpan di urutan terakhir pada dua daftar *path* yaitu *global path* dan *canvas path* milik pengguna tersebut.

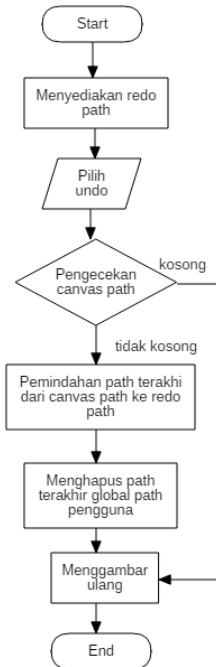
Shareable canvas pada pengguna akan menggambar ulang setiap ada penambahan *path* di *global path*.



Gambar 3.19 Diagram Alir Penyimpanan Data pada *Shareable Canvas*

3.2.3.5 Alur Perpindahan Data *Undo* dan *Redo* pada *Shareable Canvas*

Alur perpindahan data *undo* dan *redo* pada *shareable canvas* dapat dilihat pada Gambar 3.20 dan Gambar 3.21 dengan melalui beberapa tahapan.

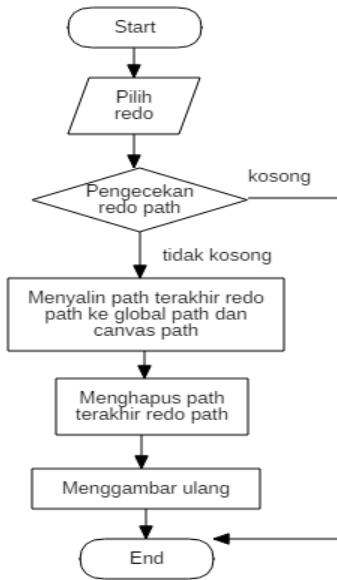


Gambar 3.20 Diagram Alir Perpindahan Data *Undo* pada *Shareable Canvas*

Tahapan-tahapan untuk satu kali penggunaan fitur *undo* pada *shareable canvas* adalah sebagai berikut:

1. Aplikasi menyediakan *redo path* untuk masing-masing pengguna.

2. Ketika pengguna menggunakan fungsi *undo*, dilakukan pengecekan pada *canvas path*.
3. Jika *canvas path* tidak kosong, maka *path* terakhir dalam *canvas path* milik pengguna tersebut dipindahkan ke akhir *redo path*.
4. Aplikasi menghapus *path* terakhir pada daftar *global path* dari pengguna yang menggunakan fitur *undo*.
5. Aplikasi menggambar ulang sesuai dengan *global path*.



Gambar 3.21 Diagram Alir Perpindahan Data Redo pada Shareable Canvas

Tahapan-tahapan untuk satu kali penggunaan fitur *redo* pada *shareable canvas* adalah sebagai berikut:

1. Aplikasi melakukan pengecekan pada *redo path* milik pengguna yang menggunakan fitur *undo*.

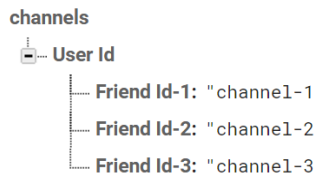
2. Apabila *redo path* milik pengguna tersebut kosong, maka fitur tidak bisa digunakan.
3. Jika *redo path* tidak kosong, maka *path* terakhir dalam *redo path* disalin dan diletakkan di akhir kedua daftar *path* yaitu *global path* dan *canvas path* milik pengguna tersebut.
4. Setelah itu, *path* terakhir dari *redo path* tersebut dihapus.
5. *Shareable canvas* menggambar ulang sesuai *global path*.

3.2.4 Perancangan Basis Data

Perancangan basis data digunakan untuk menyusun struktur data sistem pada tugas akhir ini dengan mengimplementasikan Firebase Database sebagai media penyimpanan data. Firebase Database merupakan basis data non-SQL sehingga tidak memerlukan aturan tertentu untuk mengatur struktur basis data. Oleh karena itu, rancangan basis data pada Firebase Database disimpan dalam format objek JSON *tree* yang tidak memiliki tabel atau baris seperti data SQL. Pada JSON *tree* dikenal dengan istilah *node* untuk setiap proses memasukkan data ke Firebase Database.

Terdapat beberapa *node* utama yang didalamnya terdiri dari beberapa data. Data tersebut diibaratkan sebagai baris pada basis data SQL. *Node-node* tersebut dijelaskan pada sub bab berikut:

3.2.4.1 Channels

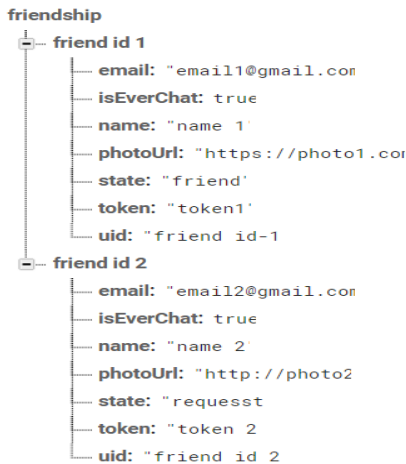


Gambar 3.22 Struktur Basis Data *Node Channels*

Node ini berisi *unique string* sebagai saluran untuk masing-masing pengguna dalam melakukan *chat* dengan pengguna lain yang dapat dilihat pada Gambar 3.22. Adapun penjelasan atribut Channels adalah sebagai berikut:

- Channels merupakan *node* utama.
- User Id adalah ID masing-masing pengguna yang digunakan sebagai *parent node* untuk daftar saluran pengguna tersebut. *Node* ini berisi daftar saluran untuk setiap teman dari pengguna tersebut.
- Friend Id adalah id teman dari *node* pengguna yang berisi saluran untuk melakukan *chat* antara pengguna dengan friend id tersebut.

3.2.4.2 Friendship



Gambar 3.23 Struktur Basis Data *Node Friendship*

Node ini berisi daftar pertemanan untuk masing-masing pengguna. Struktur *node Friendship* dapat dilihat pada Gambar 3.23 dengan penjelasan atribut sebagai berikut:

- Friendship merupakan *node* utama.
- User id adalah ID masing-masing pengguna yang digunakan sebagai *parent node* untuk daftar teman pengguna tersebut.
- Friend Id adalah ID teman dari *node* pengguna yang berisi detail pengguna yang dilihat pada Tabel 3.9.

Tabel 3.9 Detail Atribut Friend Id Node Friendship

Detail Atribut	Tipe Data	Keterangan
Email	String	Alamat <i>email</i> pengguna.
Iseverchat	Boolean	Status <i>chat</i> .
Name	String	Nama pengguna.
Photourl	String	Alamat URL foto.
State	String	Status pertemanan.
Token	String	Token pengguna.
Uid	String	Id pengguna.

3.2.4.3 Messages



Gambar 3.24 Struktur Basis Data Node Messages

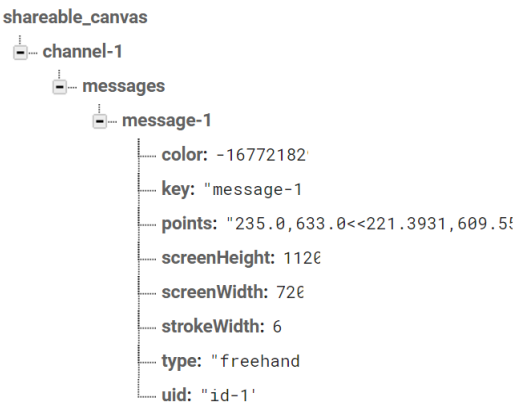
Node ini berisi daftar pesan antara pengguna yang melakukan *chat* dengan mengacu pada saluran dari *node* Channels. Struktur *node* Messages dapat dilihat pada Gambar 3.24 dengan penjelasan atribut sebagai berikut:

- Messages merupakan *node* utama.
- Channel Id merupakan saluran yang menghubungkan antara kedua pengguna dan berisi daftar pesan dari keduanya.
- Message Id merupakan ID unik dari setiap pesan yang berisi detail pesan yang dapat dilihat pada dari salah satu pengguna.

Tabel 3.10 Detail Atribut Message Id *Node* Messages

Detail Atribut	Type Data	Keterangan
Message	String	Pesan
Sender	String	Pengirim
Time	Long	Waktu pengiriman
Type	String	Tipe pesan

3.2.4.4 Shareable Canvas



Gambar 3.25 Struktur Basis Data *Node* pada Shareable Canvas

Node ini berisi data-data untuk aktivitas dalam *shareable canvas* oleh pengguna. Struktur *node* Shareable Canvas dapat dilihat pada Gambar 3.25 dengan penjelasan atribut sebagai berikut:

- Shareable Canvas merupakan *node* utama.
- Channel Id merupakan saluran yang menghubungkan antara kedua pengguna .
- Messages adalah *node* yang berisi data-data Shareable Canvas kedua pengguna.
- Messages id merupakan ID unik untuk setiap data *shareable canvas* yang berisi detail data yang dapat dilihat pada Tabel 3.11.

Tabel 3.11 Detail Atribut Messages Id Node Shareable Canvas

Detail Atribut	Tipe Data	Keterangan
Color	String	Warna
Key	String	Id pesan
Points	String	Titik koordinat
Screenheight	Int	Panjang layar
Screenwidth	Int	Lebar layar
Strokewidth	Int	Ketebalan titik
Type	String	Tipe gesture
Uid	String	Id pengirim

3.2.4.5 User Detail

Node ini berisi detail semua pengguna yang pernah masuk (*logged in*) ke dalam sistem. Struktur *node* User Detail dapat dilihat pada Gambar 3.26 dengan penjelasan atribut sebagai berikut:



Gambar 3.26 Struktur Basis Data *Node* User Detail

- User detail merupakan *node* utama.
- User Id merupakan ID unik masing-masing pengguna yang berisi detail yang dapat dilihat pada Tabel 3.12 dari pengguna tersebut.

Tabel 3.12 Detail Atribut User Id *Node* User Detail

Detail Atribut	Tipe Data	Keterangan
Email	String	Alamat <i>email</i> pengguna.
Name	String	Nama pengguna.
Photourl	String	Alamat URL foto pengguna.
Token	String	Token perangkat pengguna.
Uid	String	Id pengguna.

3.2.5 Perancangan Antarmuka

Perancangan antarmuka pengguna berisi rancangan tampilan dari aplikasi. Tampilan dirancang menggunakan Android Studio dengan format XML. Pada dasarnya terdapat empat halaman utama pada aplikasi yaitu halaman *log in*, halaman utama, halaman *chat*, dan tampilan kanvas.

3.2.5.1 Halaman *Log in*



Gambar 3.27 Halaman *Log in*

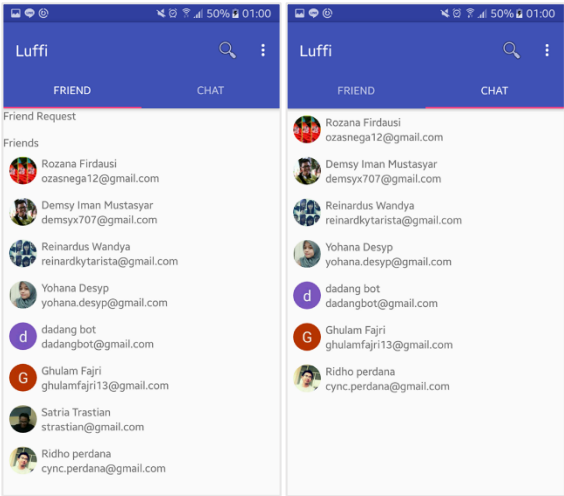
Halaman ini merupakan halaman awal apabila pengguna baru pertama kali menggunakan aplikasi ini yang dapat dilihat pada Gambar 3.27. Pada halaman ini terdapat sebuah tombol yang berfungsi untuk masuk ke halaman utama aplikasi menggunakan akun Gmail pengguna. Tombol ini secara otomatis menampilkan daftar akun Gmail yang tersinkronisasi oleh perangkat Android pengguna.

Spesifikasi atribut antarmuka untuk halaman *log in* dapat dilihat pada Tabel 3.13.

Tabel 3.13 Spesifikasi Tampilan Halaman *Log in*

Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
SignInButton	Button	Menampilkan pilihan akun Gmail yang digunakan untuk masuk ke sistem.
Title_log in	TextView	Menampilkan nama aplikasi
logoImage	CircleImageView	Menampilkan logo aplikasi

3.2.5.2 Halaman Utama



Gambar 3.28 Halaman Utama

Halaman ini merupakan halaman utama ketika pengguna telah melakukan proses *log in* yang dapat dilihat pada Gambar 3.28. Pada halaman ini terdapat dua *fragment* yaitu *fragment* Friend dan Chat. *Fragment* Friend menampilkan daftar teman

dan daftar permintaan pertemanan, sedangkan *fragment* Chat menampilkan daftar teman yang pernah melakukan *chat* dengan pengguna.

Selain itu juga terdapat tombol cari dan tombol *sign out* di bagian kanan atas tampilan. Tombol cari akan membuka halaman baru untuk mencari pengguna lain, sedangkan tombol *sign out* untuk melakukan proses *sign out* dari akun yang sedang aktif atau keluar dari sistem yang terautentikasi.

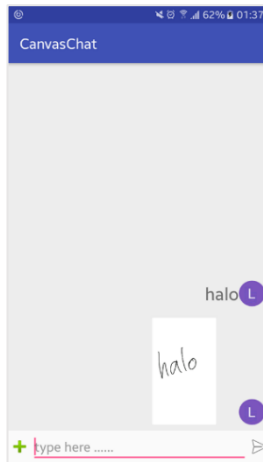
Pada bagian kiri atas tampilan halaman utama dituliskan nama pengguna yang sedang aktif. Tampilan ini digunakan untuk mengetahui akun pengguna yang sedang aktif saat ini.

Spesifikasi antarmuka halaman utama dapat dilihat pada Tabel 3.14.

Tabel 3.14 Spesifikasi Halaman Utama

Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
Listrequest	RecyclerView	Menampilkan daftar permintaan pertemanan.
Listfriend	RecyclerView	Menampilkan daftar teman.
Listchat	RecyclerView	Menampilkan daftar teman yang pernah melakukan <i>chat</i> dengan pengguna.
Searchbtn	Button	Pindah ke halaman cari pengguna
Signoutbtn	Button	Melakukan proses <i>sign out</i> akun yang sedang aktif dan menuju halaman <i>log in</i> .

3.2.5.3 Halaman *Chat*



Gambar 3.29 Halaman *Chat*

Halaman ini merupakan halaman yang menampilkan daftar pesan dari percakapan pengguna dengan salah satu pengguna lain yang dapat dilihat pada Gambar 3.29. Terdapat RecyclerView yang berfungsi menyusun pesan terurut sesuai dengan waktu pengiriman. Pada halaman ini juga terdapat kolom untuk mengisi pesan teks, tombol kirim pesan teks, dan tombol mengirim pesan dari kanvas. Spesifikasi Atribut tampilan halaman *chat* dapat dilihat pada Tabel 3.15.

Tabel 3.15 Spesifikasi Halaman *Chat*

Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
Rv_listMessage	RecyclerView	Menampilkan daftar pesan.
SendText_btn	Button	Mengirim pesan teks
TeksMessage_et	EditText	<i>Field</i> untuk menulis pesan.
CreateCanvas_btn	Button	Menampilan kanvas.

3.2.5.4 Halaman Kanvas



Gambar 3.30 Halaman Kanvas

Halaman ini merupakan halaman yang menampilkan kanvas untuk pengguna menggambar atau memberikan sentuhan seperti pada Gambar 3.30. Terdapat beberapa tombol dan satu kanvas pada halaman ini. Spesifikasi tampilan halaman kanvas dapat dilihat pada Tabel 3.16.

Tabel 3.16 Spesifikasi Halaman Kanvas

Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
LineBtn	Button	Menggunakan fitur <i>line tool</i> .
RectangleBtn	Button	Menggunakan fitur <i>rectangle tool</i> .
CircleBtn	Button	Menggunakan fitur <i>circle tool</i> .
FreehandBtn	Button	Menggunakan fitur <i>freehand tool</i> .

Tabel 3.16 Spesifikasi Halaman Kanvas

Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
Undo_btn	Button	Membatalkan grafis terakhir yang dibuat.
Redo_btn	Button	Membatalkan fungsi <i>undo</i> .
Black_btn	Button	Mengganti warna goresan menjadi warna hitam.
Red_btn	Button	Mengganti warna goresan menjadi warna merah.
Yellow_btn	Button	Mengganti warna goresan menjadi warna kuning.
ColorPicker_btn	Button	Menampilkan jendela pilihan warna.

3.2.5.5 Halaman Pencarian

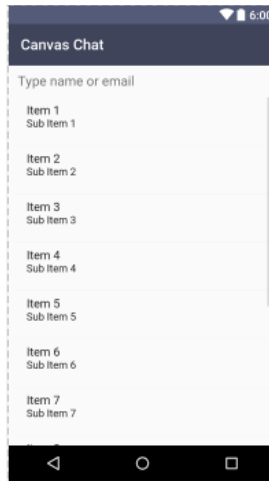
Halaman ini merupakan halaman yang menampilkan daftar pengguna aplikasi sehingga pengguna bisa mencari akun pengguna lainnya. Rancangan antarmuka halaman ini terdapat pada Gambar 3.31. Pada halaman ini terdapat kolom untuk mengisi nama atau *email* dari pengguna yang dicari.

Jika pengguna memasukkan nama atau *email* pengguna yang dicari, maka daftar pengguna disaring sehingga isi dari daftar hanya pengguna yang memiliki kesamaan nama atau *email* dari kata kunci tersebut.

Spesifikasi tampilan halaman pencarian dapat dilihat pada Tabel 3.17.

Tabel 3.17. Spesifikasi Halaman Pencarian

Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
Lv_listUser	RecyclerView	Menampilkan daftar pengguna
Keyword_et	Button	Kolom untuk memasukkan nama dan <i>email</i> .

**Gambar 3.31. Halaman Pencarian**

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi perangkat lunak sesuai analisis dan perancangan sistem secara umum pada aplikasi yang dibangun pada tugas akhir ini. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, kode sumber utama, dan implementasi antarmuka perangkat lunak.

4.1 Lingkungan Implementasi

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam membangun aplikasi pada tugas akhir ini adalah sebagai berikut:

Jenis : Laptop
Tipe : DELL Inspiron 3443
Prosesor : Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz
(4 CPUs), ~2.4GHz
Memori : 8192MB RAM
Sistem Operasi : Windows 10 Pro 64-Bit

4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun aplikasi pada tugas akhir ini adalah sebagai berikut:

- Android Studio 2.2 digunakan sebagai IDE untuk membuat sebuah *project* Android
- StarUML digunakan untuk membuat diagram kasus penggunaan dan diagram aliran.
- Postman digunakan untuk melakukan percobaan dalam memanggil suatu API.

- CorelDraw X7 digunakan untuk membuat desain aset pada aplikasi.

4.2 Implementasi Alur Kasus Penggunaan

Pada tahap ini dijelaskan implementasi dari rancangan alur kasus penggunaan. Implementasi ini dibuat dengan mengacu kepada spesifikasi dan diagram aktivitas untuk setiap kasus penggunaan.

4.2.1 *Log in*

Pada tahap ini, dijelaskan implementasi dari kasus penggunaan *log in*. Implementasi halaman *log in* yang mampu menampilkan jendela baru agar pengguna bisa menggunakan akun Gmail untuk masuk ke sistem. Implementasi untuk menyiapkan halaman *log in* pada aplikasi dapat dilihat pada Kode Sumber 4.1.

```

1. public class LoginActivity extends AppCompatActivity
2. {
3.     @Override
4.     protected void onCreate(Bundle savedInstanceState)
5.     {
6.         super.onCreate(savedInstanceState);
7.         setContentView(R.layout.activity_login);
8.         SignInButton signInButton = (SignInButton)findViewById(R.id.signInButton);
9.         signInButton.setOnClickListener(signIn());
10.    }
11.    private void signIn(){
12.        Intent loginIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
13.    }

```

Kode Sumber 4.1 Menampilkan Halaman *Log in*

Setelah mempersiapkan halaman *log in*, diimplementasikan juga cara aplikasi dalam menampilkan tampilan autentikasi menggunakan akun Gmail. Implementasi fungsi untuk autentikasi menggunakan akun Gmail dapat dilihat pada Kode Sumber 4.2.

```

1. private void firebaseAuthWithGoogleAccount(GoogleSignInAccount account) {
2.     AuthCredential credential = GoogleAuthProvider
3.         .getCredential(account.getIdToken(),null);
4.     mFirebaseAuth.signInWithCredential(credential)
5.         .addOnCompleteListener();
6. }

```

Kode Sumber 4.2 Autentikasi Menggunakan Akun Gmail

Apabila pengguna telah memilih akun Gmail yang digunakan untuk proses autentikasi. Aplikasi perlu melakukan penyimpanan detail pengguna ke dalam Firebase Database. Kemudian aplikasi mengarahkan pengguna ke halaman utama. Implementasi tahapan ini dapat dilihat pada Kode Sumber 4.3.

```

1. private void insertNewUser(){
2.     databaseReference.child("user_detail")
3.         .addListenerForSingleValueEvent(new ValueEventListener() {
4.             @Override
5.             public void onDataChange(DataSnapshot dataSnapshot) {
6.                 firebaseUser= getCurrentUser();
7.                 databaseReference.child("user_detail")
8.                     .child(firebaseUser.getId())
9.                     .setValue(firebaseUser);
10.                startActivity(new Intent(LoginActivity.this
11.                    ,MainActivity.class));
12.                finish();
13. }

```

Kode Sumber 4.3 Menambahkan Detail Pengguna

4.2.2 Mengirim Pesan

Pada tahap ini, dijelaskan implementasi dari kasus penggunaan mengirim pesan. Pengguna dihadapkan dengan halaman *chat* yang mampu menampilkan media *input* kanvas dan teks. Implementasi untuk menampilkan media *input* kanvas dilihat pada Kode Sumber 4.4.

```

1. private void popupCanvas(){
2.     final PopupWindow popupWindow;
3.     LayoutInflater inflater = getSystemService(C
        ontext.LAYOUT_INFLATER_SERVICE);
4.     View layout = inflater.inflate(R.layout.crea
        te_canvas,
5.     (ViewGroup)findViewById(R.id.popup_canvas));
6.     popupWindow = new PopupWindow(layout);
7.     popupWindow.showAtLocation(CENTER,0,0);
8.     popupWindow.setOutsideTouchable(true);
9.     popupWindow.setFocusable(true);
10.    offlineCanvasHolder = new canvasHolder(layout);
11.    canvasController = new DrawView(this);
12.    canvasHolder.canvas.addView(canvasControl);
13.    final View mView = canvasHolder.canvas;
14. }

```

Kode Sumber 4.4 Menampilkan Media *Input* Kanvas

Tahapan awal pengiriman adalah mengonversi kanvas dalam bentuk gambar. Setelah tahapan konversi, aplikasi mengunggah gambar tersebut ke Firebase Storage. Gambar diletakkan pada masing-masing folder milik pengguna pada Firebase Storage. Dari semua pengunggahan gambar yang telah dilakukan, ukuran gambar berada pada kisaran 30 hingga 160 kilobyte, dipengaruhi oleh jumlah data berupa goresan yang ada pada kanvas. Implementasi tahap mengunggah gambar dapat dilihat pada Kode Sumber 4.5.

```

1. private void uploadCanvas(View view, final PopupWind
   ow popupWindow){
2.     StorageReference imageDir = storageReference
3.     .child("image");
4.     StorageReference imageRef = imageDir
5.     .child(firebaseUser.getUid())
6.     .child(+Calendar.getInstance()
7.     .getTimeInMillis()+".jpg");
8.     Bitmap b = Bitmap.createBitmap();
9.     Canvas c = new Canvas(b);
10.    view.layout();
11.    view.draw(c);
12.    ByteArrayOutputStream baos = new
13.        ByteArrayOutputStream();
14.        b.compress(Bitmap.CompressFormat.JPEG,100,baos);

15.    byte[]data = baos.toByteArray();
16.    uploadTask = imageRef.putBytes(data);
17.    addOnSuccessListener({
18.        @Override
19.        public void onSuccess(taskSnapshot) {
20.            Uri uri = taskSnapshot.getDownloadUrl();
21.            .....
22.        });
23. }

```

Kode Sumber 4.5 Konversi Kanvas ke Gambar dan Proses Unggah

Proses unggah sebuah gambar ke Firebase Storage menghasilkan URL sebagai tautan untuk mengakses gambar tersebut melalui akses internet. Tautan ini dikirim ke lawan *chat* melalui Firebase Database sehingga pengguna lawan bisa menerima pesan. Tahapan pengiriman URL diimplementasikan seperti pada Kode Sumber 4.6.

```

1. HashMap<String,Object> message = new HashMap<>();
2. message.put("message",uri.toString());
3. message.put("type","image");
4. message.put("time", ServerValue.TIMESTAMP);
5. message.put("sender",firebaseUser.getDisplayName());

6. databaseReference.child("messages")
7. .child(channel_id).push()
8. .setValue(message);

```

Kode Sumber 4.6 Mengirim Pesan Berisi Alamat Gambar

Selain mengirim pesan berupa gambar, pengguna juga diberi pilihan untuk mengirim pesan berupa teks. Sistem menyediakan kolom teks sehingga pengguna bisa menuliskan pesannya di kolom tersebut. Pengiriman pesan teks diimplementasikan seperti pada Kode Sumber 4.7.

```

1. HashMap<String, Object> newMessage =
2.     new HashMap<>();
3. newMessage.put("message",textMessage_et
4.     .getText().toString());
5. newMessage.put("type","text");
6. newMessage.put("time", ServerValue.TIMESTAMP);
7. newMessage.put("sender",firebaseUser
8.     .getDisplayName());
9. databaseReference.child("messages")
10. .child(channel_id).push()
11. .setValue(newMessage);

```

Kode Sumber 4.7 Mengirim Pesan Teks

Proses ini mengakhiri tahapan pada sisi pengirim. Kemudian dilanjutkan dengan proses menerima pesan oleh pengguna. Aplikasi pada sisi penerima mengakses Firebase Storage sesuai dengan URL yang diterima dan disimpan dalam

penyimpanan lokal. Proses ini dapat dilihat pada Kode Sumber 4.8.

```

1. ListMessageAdapter listMessageAdapter;
2. private void setAdapter(){
3. listMessageAdapter = new ListMessageAdapter
4. (Message.class, R.layout.item_message,
5. MessageHolder.class, databaseReference
6. .child("messages").child(channel_id)
7. ,receiver.getPhotoUrl(),this);
8. }
9. protected void populateViewHolder () {
10. Glide.with(activity)
11. .load(model.getMessage())
12. .into(viewHolder.imageMessage);
13. }

```

Kode Sumber 4.8 Menerima Pesan dan Menyimpan di Penyimpanan Lokal

4.2.3 Berbagi Kanvas

Pada tahap ini, dijelaskan implementasi dari kasus penggunaan mengirim pesan. Sistem menampilkan sebuah halaman kanvas yang terhubung dengan salah satu pengguna lain yang telah dipilih sebelumnya.

Pertama sistem menyediakan sebuah saluran yang digunakan untuk menghubungkan dua pengguna. Apabila sebelumnya kedua pengguna pernah terhubung, maka tidak perlu dilakukan pembuatan saluran baru. Namun jika kedua pengguna tidak pernah terhubung, maka saluran baru perlu dibuat. Saluran ini dibuat dari penggabungan ID kedua pengguna yang terhubung. Saluran yang dibuat disimpan di Firebase Database pada *node* Channels.

Proses membuat suatu saluran diimplementasikan seperti Kode Sumber 4.9.

```

1. private void getChannel(){
2.     databaseReference.child("channels/"
3.         +firebaseUser.getUid())
4.         .addListenerForSingleValueEvent(
5.             new ValueEventListener() {
6.                 @Override
7.                 public void onDataChange() {
8.                     if(database.hasChild(receiver.getUid()))
9.                     {
10.                        channel_id = dataSnapshot.child(
11.                            receiver.getUid())
12.                            .getValue().toString()+"_sc";
13.                    }
14.                else {
15.                    HashMap<String , Object> selfChannel =
16.                        new HashMap<>();
17.                    selfChannel.put(receiver.getUid(),
18.                        firebaseUser.getUid()+
19.                        receiver.getUid());
20.                    databaseReference.child("channels/")
21.                        .child(firebaseUser.getUid())
22.                        .updateChildren(selfChannel);
23.                }

```

Kode Sumber 4.9 Inisialisasi Saluran *Shareable Canvas*

Setelah dibuat suatu saluran yang menghubungkan kedua pengguna, barulah kedua pengguna bisa menggunakan kanvas. Apabila salah satu pengguna membuat sebuah goresan, maka secara otomatis kanvas pengguna lainnya akan diperbarui.

Selama proses perpindahan sentuhan di kanvas, aplikasi menyimpan titik koordinat sebagai data yang dikirim ke pengguna lain. Proses pengiriman baru dilakukan apabila pengguna melepaskan sentuhan dari kanvas.

Tahapan selanjutnya adalah menangani setiap sentuhan yang diimplementasikan seperti pada Kode Sumber 4.10.

```

1. private Path mPath;
2. @Override
3. public boolean onTouch(View view, MotionEvent event)
4. {
5.     if(PaintTool.equals("freehand"))
6.     {
7.         switch (event.getAction()) {
8.             case MotionEvent.ACTION_DOWN:
9.                 drawingStart();
10.                invalidate();
11.                break;
12.            case MotionEvent.ACTION_MOVE:
13.                drawingMove(x, y);
14.                invalidate();
15.                break;
16.            case MotionEvent.ACTION_UP:
17.                sendNewData(drawingFinish(mPath));
18.                invalidate();
19.        }

```

Kode Sumber 4.10 Menangani Setiap Jenis Sentuhan

Setelah pengguna melepaskan sentuhan dari layar, grafis yang telah dibuat disimpan ke dalam Firebase Database. Sebelum menyimpan data pada Firebase Database, aplikasi perlu menyediakan suatu objek berisi atribut yang telah dirancang pada perancangan basis data, yaitu pada *node* Shareable Canvas.

Salah satu data utama penting yang harus disimpan adalah titik-titik koordinat dari goresan yang pengguna buat. Titik tersebut tersimpan dalam variabel *path* yang disediakan oleh Android Graphics Library. Proses ini dapat dilihat pada Kode Sumber 4.11.

```

1. private ShareableItem drawFinish(){
2.     Path newPath = new Path();
3.     newPath.drawObject();
4.     ShareableItem item =
5.     new ShareableItem(newPath);
6.     return item;
7. }
8. private void sendNewData(ShareableItem item){
9.     databaseReference.child("shareable_canvas")
10.     .child(channel_id).child("messages")
11.     .child(item.getKey()).setValue(item);
12. }

```

Kode Sumber 4.11 Mengirim Path *Shareable Canvas*

Apabila ada data baru yang diterima oleh Firebase Database, secara otomatis akan memberikan pemberitahuan kepada lawan *chat* sehingga kanvas pada lawan *chat* diperbarui. Proses ini dapat dilihat pada Kode Sumber 4.12.

```

1. public ShareableCanvasView(Context context,Stringchannel_id){
2.     databaseReference.child("shareable_canvas"+
3.     "/channel_id+"/messages")
4.     .addChildEventListener(... {
5.     public void onChildAdded(...) {
6.         ShareableItem newItem = dataSnapshot
7.         .getValue(ShareableItem.class);
8.         addPathFromDatabase(newItem);
9.     }
10. }
11. private void addPathFromDatabase
12. (ShareableItem newItem) {
13.     mCanvas.drawPath(newItem.getPath(),
14.     newItem.getShareablePaint().getPaint());
15.     paths.add(newItem);
16.     referencePath.put(newItem.getKey(),newItem;
17. }

```

Kode Sumber 4.12 Mengirim Path Pada *Sharable Canvas*

4.2.4 Mencari Akun Pengguna Lain

Pada tahap ini, dijelaskan implementasi dari kasus penggunaan mencari akun pengguna lain. Sistem menampilkan sebuah halaman pencarian sehingga pengguna bisa memasukkan nama atau email dari pengguna yang dicari.

Tahapan pertama yaitu sistem mengambil semua data pengguna yang ada di Firebase Database. Semua data yang diterima disimpan dalam sebuah variabel sementara untuk dicocokkan sesuai nama atau email yang pengguna masukkan pada kolom pencarian. Tahapan mengambil semua data pengguna diimplementasikan seperti pada Kode Sumber 4.13.

```
1. private void getAllUser(){  
2.     DatabaseReference reference =  
3.         FirebaseDatabase.getInstance()  
4.         .getReference("user_detail");  
5.     reference.addListenerForSingleValueEvent(  
6.         .....  
7.         for (DataSnapshot child : dataSnapshot.getChildren  
8.             ()){  
9.                 listUser.add(child.getValue(User.class));  
10.                adapter.notifyDataSetChanged();  
11.            }
```

Kode Sumber 4.13 Kode Mengambil Semua Data Pengguna

Setelah semua data diterima, data tersebut dijadikan data utama yang disaring sesuai nama atau email yang dimasukkan oleh pengguna pada kolom pencarian. Penyaringan dilakukan setiap ada satu karakter yang dimasukkan ke dalam kolom pencarian.

Proses penyaringan data pengguna dapat dilihat pada Kode Sumber 4.14.

```

1. @Override
2. protected FilterResults (CharSequence constraint) {
3.     FilterResults results = new FilterResults();
4.     if(constraint==null
5.         ||constraint.length()==0){
6.         results.values = allData;
7.         results.count = allData.size();
8.     }
9.     else {
10.        ArrayList<User> resultFilter;
11.        for (User data : allData){
12.            if(data.getName().toLowerCase()
13.                .contains(constraint)
14.                ||data.getEmail().toLowerCase()
15.                .contains(constraint)){
16.                resultFilter.add(data);
17.            }
18.        }
19.        results.values = resultFilter;
20.        results.count = resultFilter.size();
21.    }
22.    return results;
23. }
24.
25. void publishResults(constraint, results) {
26.    filteredData = results.values;
27.    notifyDataSetChanged();
28. }

```

Kode Sumber 4.14 Menyaring Data Pengguna

Data yang telah disaring inilah yang ditampilkan ke pengguna sehingga pengguna menemukan akun pengguna lain yang memiliki kriteria *email* atau nama sesuai dengan isi dari kolom pencarian.

4.2.5 Menambahkan Teman

Pada tahap ini, dijelaskan implementasi dari kasus penggunaan menambahkan teman. Sistem menampilkan jendela baru pada halaman pencarian akun pengguna lain. Jendela baru yang ditampilkan berisi detail dan tombol untuk mengirim permintaan pertemanan. Implementasi untuk menampilkan jendela ini dapat dilihat pada Kode Sumber 4.15.

```

1. private void initiatePopUpWindow(final User user, Vi
   ew view){
2.     final PopupWindow popupWindow;
3.     LayoutInflater layoutInflater =
4.     (LayoutInflater)
5.     .getSystemService(LAYOUT_INFLATER_SERVICE);
6.     View layout = layoutInflater
7.         .inflate(R.layout.user_preview,
8.             (ViewGroup)view
9.             .findViewById(R.id.preview_layout));
10.    int width = view.getWidth()*6/7;
11.    int height = view.getHeight()*6/7;
12.    popupWindow = new PopupWindow();
13.    popupWindow.showAtLocation(view);
14.    PreviewHolder previewHolder =
15.        new PreviewHolder();
16.    previewHolder.name_tv.setText(user.getName());
17.    previewHolder.email.setText(user.getEmail());
18. }

```

Kode Sumber 4.15 Menampilkan Jendela Konfirmasi Permintaan Pertemanan

Setelah menampilkan sebuah jendela baru, pengguna bisa mengirim permintaan pertemanan dengan menekan tombol Add Tombol ini melakukan pengubahan data pada Firebase Database berupa status pertemanan. Implementasi tahapan ini dapat dilihat pada Kode Sumber 4.16.

```

1. private void addFriend(final User friend){
2.     friend.setState("friend");
3.     databaseReference.child("friendship/"+
4.     mUser.getId()+"/"+
5.     friend.getId()).setValue(friend)
6.     .addOnSuccessListener({
7.         @Override
8.         public void onSuccess(Void aVoid) {
9.             if(listFriend==null){
10.                listFriend = new HashMap<>();
11.            }
12.            listFriend.put(friend.getId(),friend);
13.        }
14.    });
15.    mUser.setState("request");
16.    databaseReference.child("friendship/"
17.    +friend.getId()+"/"+mUser.getId())
18.    .setValue(mUser);
19. }

```

Kode Sumber 4.16 Mengirim Permintaan Pertemanan

4.2.6 Mengonfirmasi Permintaan Pertemanan

Pada tahap ini, dijelaskan implementasi dari kasus penggunaan mengonfirmasi permintaan pertemanan. Sistem menampilkan semua permintaan pertemanan di halaman utama aplikasi. Pengguna bisa memilih permintaan pertemanan yang ingin dikonfirmasi dari daftar permintaan pertemanan.

Apabila pengguna memilih salah satu permintaan pertemanan, maka sistem menampilkan detail permintaan pertemanan dan sebuah tombol Add untuk mengonfirmasi permintaan pertemanan.

Implementasi untuk menampilkan jendela konfirmasi permintaan pertemanan dapat dilihat pada Kode Sumber 4.17.


```

19. void initiatePopUpWindow(User user, View view){
20.     final PopupWindow popupWindow;
21.     LayoutInflater inflater =
22.         (LayoutInflater)
23.             .getSystemService(LAYOUT_INFLATER_SERVICE);
24.     View layout = inflater
25.         .inflate(R.layout.user_preview,
26.             (ViewGroup)view
27.             .findViewById(R.id.preview_layout));
28.     int width = view.getWidth()*6/7;
29.     int height = view.getHeight()*6/7;
30.     popupWindow = new PopupWindow();
31.     popupWindow.showAtLocation(view);
32.     PreviewHolder previewHolder =
33.         new PreviewHolder();
34.     previewHolder.name_tv.setText(user.getName());
35.     previewHolder.email_tv.setText(user.getEmail);
36. }

```

Kode Sumber 4.17 Menampilkan Jendela Konfirmasi Permintaan Pertemanan

Setelah menampilkan jendela konfirmasi pertemanan, proses mengonfirmasi permintaan pertemanan yang diimplementasikan seperti pada Kode Sumber 4.18.

```

1. private void addFriend(final User friend){
2.     friend.setState("friend");
3.     databaseReference.child("friendship/"+
4.         firebaseUser.getUid()+"/"+
5.         friend.getUid()).setValue(friend);
6. }

```

Kode Sumber 4.18 Mengonfirmasi Permintaan Pertemanan

4.3 Implementasi Alur Penyimpanan

Pada tahap ini dijelaskan implementasi dari rancangan alur penyimpanan data dalam membangun aplikasi. Implementasi pada tahap ini meliputi enam bagian sebagai berikut.

4.3.1 Implementasi Pengolahan Data Sentuhan pada Kanvas

Pada bagian ini dijelaskan implementasi pengolahan data sentuhan pada kanvas sesuai dengan rancangan yang dibuat sebelumnya.

Proses penanganan tiga jenis sentuhan yaitu sentuhan awal saat pengguna pertama kali memberi sentuhan, melakukan perpindahan sentuhan, dan saat pengguna melepas sentuhan pada kanvas. Proses ini dapat dilihat pada Kode Sumber 4.19.

```

1. public class DrawView extends View implements View.
   OnTouchListener {
2. @Override
3. public boolean onTouch(View view, MotionEvent event)
   {
4. switch (event.getAction()) {
5.     case MotionEvent.ACTION_DOWN:
6.         touch_start(x, y);
7.         invalidate();
8.         break;
9.     case MotionEvent.ACTION_MOVE:
10.        touch_move(x, y);
11.        invalidate();
12.        break;
13.    case MotionEvent.ACTION_UP:
14.        touch_up();
15.        drawPath();
16.        invalidate();
17.        break;
18.    }
19. }
20. }

```

Kode Sumber 4.19 Penanganan Sentuhan Layar

Proses menggambar suatu objek seperti persegi panjang, lingkaran, dan garis dilakukan ketika pengguna melepaskan sentuhan dari kanvas. Namun untuk objek menggambar bebas, penggambaran dilakukan setiap perpindahan pada sentuhan. Proses menggambar pada kanvas dilakukan dengan memanggil fungsi draw yang dimiliki kelas Canvas. Proses menggambar objek dapat dilihat pada Kode Sumber 4.20.

```

1. private void FreehandMove(float x, float y) {
2.   mPath.quadTo(mX, mY, (x + mX)/2, (y + mY)/2);
3.   mX = x;
4.   mY = y;
5.   mCanvas.drawPath()
6. }
7. private void drawRectangle(){
8.   newPath.addRect(points.get("left"),
9.   points.get("top"),
10.  points.get("right"),points.get("bottom"),
11.  Path.Direction.CCW);
12.  mCanvas.drawPath(newPath,mPaint);
13. }
14. private void drawCircle(){
15.  newPath.addOval(points.get("left"),
16.  points.get("top"),
17.  points.get("right"),points.get("bottom"),
18.  Path.Direction.CCW);
19.  mCanvas.drawPath(newPath,mPaint);
20. }
21. private void drawLine(){
22.  newPath.moveTo(startX,startY);
23.  newPath.lineTo(endX,endY);
24.  mCanvas.drawPath(newPath,mPaint);
25.  paths.add(Pair.create(newPath,newPaint));
26. }

```

Kode Sumber 4.20 Menggambar Objek pada Kanvas

4.3.2 Implementasi Penyimpanan Data pada *Offline Canvas*

Pada bagian ini dijelaskan implementasi penyimpanan data pada *offline canvas* sesuai dengan rancangan yang dibuat sebelumnya.

Setiap pengguna memberikan goresan dan kemudian melepaskan sentuhan dari layar. Titik koordinat pada goresan disimpan dalam variabel *path*. *Path* tersebut ditambahkan ke dalam sebuah variabel yang disebut *canvas path*, berisi daftar semua *path* yang pernah dibuat. *Canvas path* menyimpan semua *path* sesuai urutan dibuatnya. *Canvas path* merupakan inisialisasi dari kelas *ArrayList* yang memungkinkan untuk menyimpan beberapa objek secara berurutan tanpa perlu mendeklarasikan banyak variabel.

Proses ini dapat dilihat pada Kode Sumber 4.21.

```
1. public class DrawView extends View implements View.  
   OnTouchListener {  
2. private ArrayList<Pair<Path,Paint>> canvasPaths;  
3. private Path mPath;  
4. private Canvas mCanvas;  
5. private void touch_up() {  
6.     mPath.drawObject(mX, mY);  
7.     Paint newPaint = new Paint();  
8.     newPaint = mPaint;  
9.     mCanvas.drawPath(mPath, mPaint);  
10.    canvasPaths.add(Pair.create(mPath, newPaint));  
11.    mPath = new Path();  
12. }  
13. }
```

Kode Sumber 4.21 Penyimpanan Goresan dalam *Path*

4.3.3 Implementasi Alur Perpindahan Data pada Fitur *Undo* dan *Redo Offline Canvas*

Pada bagian ini dijelaskan implementasi alur perpindahan data pada fitur *undo* dan *redo offline canvas* sesuai dengan rancangan yang dibuat sebelumnya. Diperlukan sebuah objek hasil inisialisasi kelas `ArrayList` yang berfungsi untuk menyimpan daftar *path* yang dibatalkan pada fitur *undo*. Objek ini disebut *redo paths*. Sebelum melakukan proses *undo*, aplikasi melakukan pemeriksaan data kanvas. Apabila kanvas kosong atau tidak ada *path* pada *canvas path*, maka fungsi *undo* tidak bisa digunakan.

Proses inisialisasi sebuah *redo path* dan penggunaan fitur *undo* dapat dilihat pada Kode Sumber 4.22.

```

1. public class DrawView{
2.     private ArrayList redoPaths;
3.     public void onClickUndo () {
4.         if (canvasPaths.size()>0)
5.         {
6.             redoPath.add(
7.                 canvasPaths.remove(
8.                     canvasPaths.size()-1));
9.             invalidate();
10.        }
11.    }

```

Kode Sumber 4.22 Inisialisasi *Redo Path* dan Fungsi *Undo*

Apabila pengguna pernah menggunakan fungsi *redo* dan belum membuat *gesture* lain setelah itu, maka fungsi *redo* dapat digunakan. Selain itu, fungsi *redo* bisa digunakan apabila terdapat *path* dalam *redo path*. Ketika fungsi ini dipanggil, *path* terakhir pada *redo path* akan ditambahkan ke *canvas path*. Setelah berhasil diambahkan maka dilakukan penggambaran ulang kanvas sesuai data yang ada pada *canvas path*.

Proses penggunaan fitur *redo* dapat dilihat pada Kode Sumber 4.23.

```

1. public class DrawView {
2.     private ArrayList redoPaths;
3.
4.     public void onClickRedo () {
5.         if (undonePaths.size()>0)
6.         {
7.             canvasPath.add(
8.                 redoPaths.remove(redoPaths.size()-1));
9.             invalidate();
10.        }
11.    }
12. }

```

Kode Sumber 4.23 Penggunaan Fungsi *Redo*

4.3.4 Implementasi Alur penyimpanan Data pada *Shareable Canvas*

Pada bagian ini dijelaskan implementasi alur penyimpanan data pada *shareable canvas* sesuai dengan rancangan yang dibuat sebelumnya. Aplikasi menyediakan sebuah variabel yang berisi daftar *path* yang pernah dibuat oleh kedua pengguna. Daftar *path* ini disebut *global path*. Selain itu, diperlukan juga sebuah variabel yang menyimpan daftar *path* yang pernah dibuat oleh masing-masing pengguna. Daftar *path* ini disebut *my paths*.

Jika salah satu pengguna memberikan sentuhan pada kanvas, maka sentuhan tersebut disimpan dalam variabel *path*. Setelah pengguna tersebut melepaskan sentuhan, maka *path* tersebut dikirim dan disimpan ke dalam dua daftar *path* yang telah diinisialisasi, yaitu *global path* dan *my paths*.

Proses penyediaan *global path* dan penyimpanan *path* terakhir yang dibuat dapat dilihat pada Kode Sumber 4.24.

```

1. public class ShareableCanvasView extends View
2. {
3. ....
4. private ArrayList<ShareableItem> globalPaths =
5.     new ArrayList<>();
6. private ArrayList<ShareableItem> myPaths =
7.     new ArrayList<>();
8.
9. private ShareableItem touchFinish()
10. {
11.     mPath.drawObject();
12.     mCanvas.drawPath(mPath, mPaint);
13.
14.     ShareableItem item = new ShareableItem(mPath);
15.
16.     globalPaths.add(item);
17.     mPath = new Path();
18.     .....
19.     myPaths.add(item);
20.
21.     return item;
22. }
23. ....
24. }

```

Kode Sumber 4.24 Proses Penyimpanan Path pada *Shareable Canvas*

4.3.5 Implementasi Alur Perpindahan Data pada Fitur *Undo* dan *Redo Shareable Canvas*

Pada bagian ini dijelaskan alur perpindahan data pada fitur *undo* dan *redo shareable canvas* sesuai dengan rancangan yang dibuat sebelumnya. Fungsi *undo* hanya bisa dilakukan terhadap hasil sentuhan milik masing-masing pengguna. *Shareable canvas* tidak memungkinkan pengguna melakukan *undo* terhadap gambar yang dibuat pengguna lain. Sama seperti pada *offline*

canvas, diperlukan sebuah variabel dari kelas `ArrayList` untuk menyimpan daftar *path* yang dibatalkan pada kanvas. Daftar *path* ini disebut *redo paths*.

Setiap kali seorang pengguna menggunakan fitur *undo* pada *shareable canvas*, *path* terakhir yang dibuat pada oleh pengguna tersebut dihapus dari kanvas dan dipindahkan ke *redo paths*. *Path* ini berada pada *my paths* milik pengguna tersebut. Setelah *path* dipindahkan ke *redo path*, *path* serupa yang berada pada *global path* juga dihapus sehingga kanvas pada pengguna lainnya juga ikut menghapus *path* tersebut.

Proses penyediaan *redo path* dan penggunaan fitur *undo* dapat dilihat pada Kode Sumber 4.25

```

1. private ArrayList<ShareableItem> redoPaths = new
   ArrayList<>();
2. public void onClickUndo ()
3. {
4.     if (myPath.size()>0)
5.     {
6.         ShareableItem deletedPath =
7.         myPath.remove(myPath.size()-1);
8.         globalPaths.remove(deletedPath);
9.         redoPaths.add(deletedPath);
10.        deleteFirebase (
11.        deletedPath.getKey());
12.        invalidate();
13.    }
14. }
```

Kode Sumber 4.25 Inisialisasi *Redo Path* dan Fungsi *Undo* pada *Shareable Canvas*

Jika seorang pengguna pernah menggunakan fungsi *undo*, maka pengguna tersebut bisa menggunakan fungsi *redo* untuk mengembalikan *path* yang terakhir dihapus oleh fitur *undo*. Proses mengembalikan *path* terakhir dimulai dengan melakukan

pengecekan isi dari *redo paths*. Jika *redo paths* tidak kosong, maka *path* terakhir pada *redo path* kembali dimasukkan kedalam *global path* sehingga kanvas pada pengguna lainnya ikut diperbarui. *Path* ini juga ditambahkan pada *my paths* milik pengguna tersebut.

Proses penggunaan fitur *redo* dapat dilihat pada Kode Sumber 4.26.

```

1. private ArrayList<ShareableItem> redoPaths;
2. public void onClickRedo (){
3.     if (redoPaths.size()>0){
4.         ShareableItem path = redoPaths.remove(
5.             redoPaths.size()-1);
6.         globalPaths.add(path);
7.         myPath.add(path);
8.         sendNewData(path);
9.         invalidate();
10.    }
11. }

```

Kode Sumber 4.26 Proses Penggunaan Fitur *Redo* pada *Shareable Canvas*

4.4 Implementasi Basis Data

Pada tahap ini digambarkan implementasi dari desain basis data yang telah dibuat. Implementasi dibagi menjadi beberapa bagian sesuai jumlah *node* yaitu Channels, Friendship, Messages, Shareable Canvas, dan User Detail.

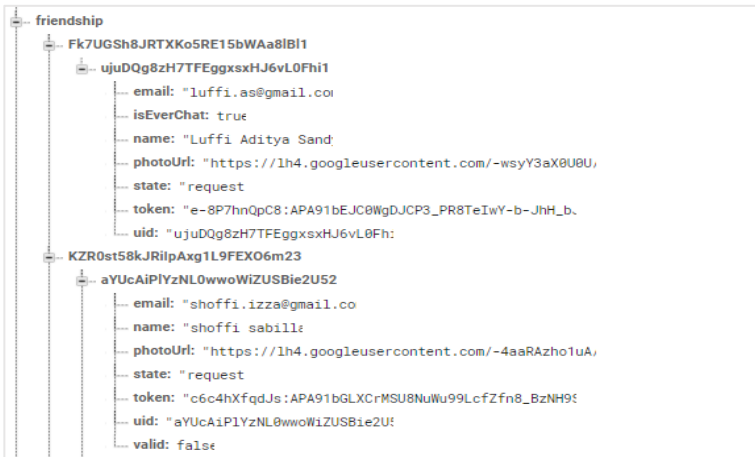
4.4.1 Implementasi *Node Channels*

Pada bagian ini ditampilkan implementasi dari *node Channels*. Implementasi dapat dilihat pada Gambar 4.1.

Gambar 4.1 Implementasi *Node Channels*

4.4.2 Implementasi *Node Friendship*

Pada bagian ini ditampilkan implementasi dari *node Friendship*. Implementasi dapat dilihat pada Gambar 4.2.

Gambar 4.2 Implementasi *Node Friendship*

4.4.3 Implementasi *Node Messages*

Pada bagian ini ditampilkan implementasi dari *node Messages*. Implementasi dapat dilihat pada Gambar 4.3.



Gambar 4.3 Implementasi *Node Messages*

4.4.4 Implementasi *Node Shareable Canvas*

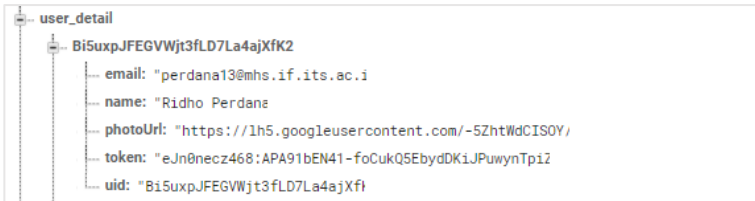
Pada bagian ini ditampilkan implementasi dari *node Shareable Canvas*. Implementasi dapat dilihat pada Gambar 4.4.



Gambar 4.4 Implementasi *Node Shareable Canvas*

4.4.5 Implementasi *Node User Detail*

Pada bagian ini ditampilkan implementasi dari *node User Detail*. Implementasi dapat dilihat pada Gambar 4.5.



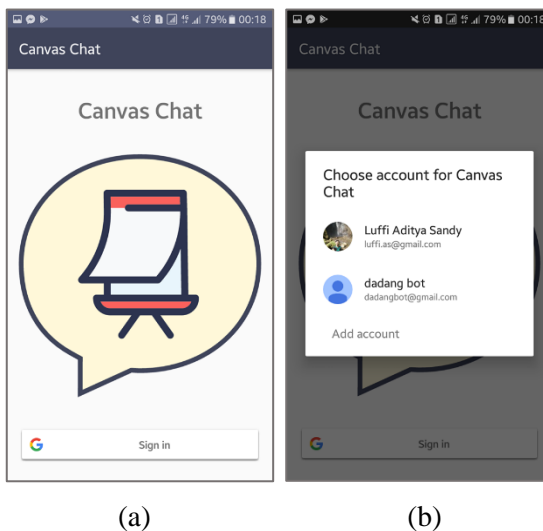
Gambar 4.5 Implementasi *Node* User Detail

4.5 Implementasi Antarmuka

Pada tahap ini dijelaskan implementasi dari antarmuka yang telah dirancang pada BAB III.

4.5.1 Halaman *Log in*

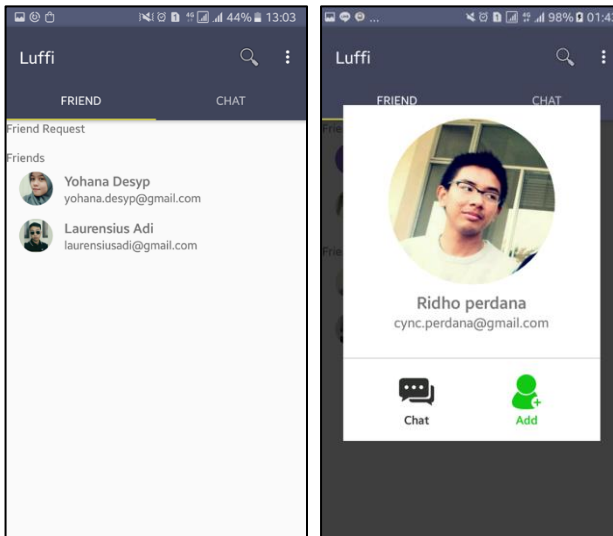
Pada halaman ini pengguna diminta untuk melakukan *sign in* dengan akun Gmail yang terintegrasi dengan perangkat Android milik pengguna tersebut. Implementasi halaman *log in* dapat dilihat pada Gambar 4.6.



Gambar 4.6 Implementasi Halaman *Log in*

4.5.2 Halaman Utama

Pada Halaman ini terdapat dua *fragment* yaitu Friend dan Chat. *Fragment* Friend menampilkan daftar teman dan permintaan pertemanan dari pengguna, sedangkan *fragment* Chat menampilkan daftar pengguna lain yang pernah melakukan *chat* dengan akun pengguna yang sedang masuk. Implementasi halaman utama dapat dilihat pada Gambar 4.7.



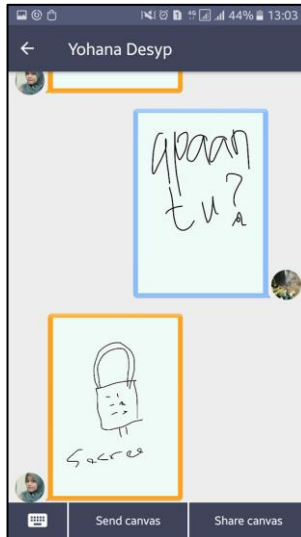
(a)

(b)

Gambar 4.7 Implementasi Halaman Utama

4.5.3 Halaman Chat

Pada halaman ini ditampilkan pesan-pesan yang pernah dikirim oleh kedua pengguna. Terdapat tiga tombol di bagian bawah halaman yaitu tombol pesan teks, kanvas, dan membagikan kanvas ke pengguna lain sebagai fitur yang disediakan aplikasi. Implementasi halaman *chat* dapat dilihat pada Gambar 4.8.

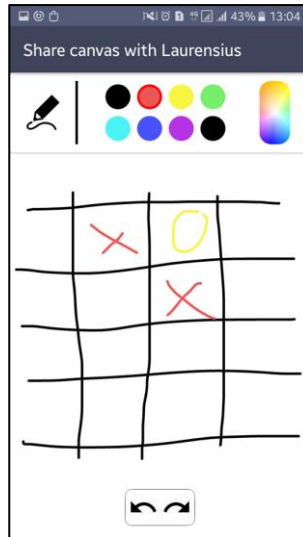


Gambar 4.8 Implementasi Halaman Chat

4.5.4 Halaman Kanvas

Pada halaman ini ditampilkan sebuah kanvas untuk membuat suatu grafis yang hasil konversi gambarnya dapat dikirimkan ke pengguna lain atau kanvas ini juga bisa menjadi sebuah kanvas yang bisa dibagikan ke pengguna lain.

Pada bagian atas terdapat pilihan gambar dan pengguna juga bisa memilih warna selain dari warna yang disediakan melalui tombol “Color Picker” pada bagian atas kanan halaman. Selain itu pengguna bisa memilih bentuk objek yang digambar bisa berupa persegi empat, lingkaran, atau garis. Selain itu terdapat tombol untuk melakukan fungsi *undo* dan *redo* pada kanvas pada bagian bawah halaman. Implementasi halaman utama dapat dilihat pada Gambar 4.9.



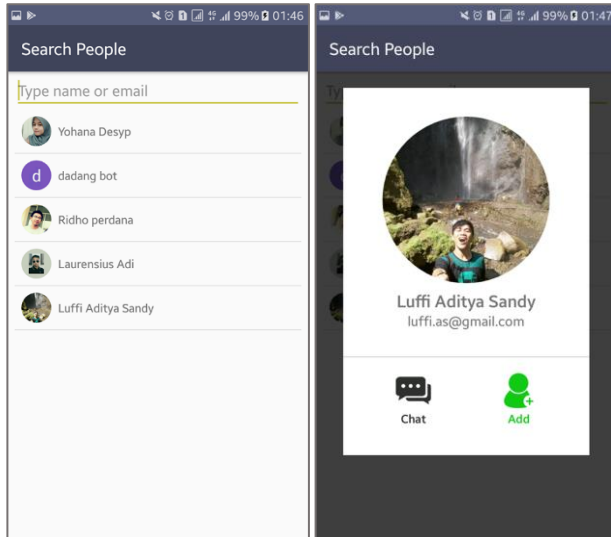
Gambar 4.9. Implementasi Halaman Kanvas

4.5.5 Halaman Pencarian

Pada halaman ini ditampilkan daftar setiap pengguna yang pernah menggunakan aplikasi. Pengguna juga bisa mencari akun pengguna lain dengan memasukkan nama atau *email* dari pengguna yang dicari. Daftar pengguna secara langsung disaring sesuai data yang dimasukkan.

Jika pengguna memilih salah satu akun dari daftar, maka aplikasi akan menampilkan sebuah jendela baru yang berisi detail dari pengguna beserta dua tombol, yaitu “add” untuk menambahkan akun menjadi teman dan tombol “chat” untuk mengirim pesan. Ketika tombol “add” ditekan, aplikasi mengirim notifikasi ke akun yang dipilih sebagai sebuah permintaan pertemanan, sedangkan jika tombol “chat” ditekan maka aplikasi mengarahkan pengguna ke halaman *chat*.

Implementasi halaman pencarian dapat dilihat pada Gambar 4.10.



(a)

(b)

Gambar 4.10. Implementasi Halaman Pencarian

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini dijelaskan tentang uji coba dan evaluasi dari implementasi yang telah dilakukan .

5.1 Lingkungan Pelaksanaan Pengujian

Lingkungan uji coba yang digunakan adalah dua perangkat *smartphone* dengan sistem operasi Android, spesifikasinya dapat dilihat di bawah ini:

1. Tipe : Samsung A320
 Prosesor : Octa-core 1.6 GHz Cortex-A53
 Memori : 2GB
 Sistem Operasi : Marshmallow
 Ukuran Layar : 4.7 inci
2. Tipe : Xiami Mi 4
 Prosesor : Quad-core 2.5 GHz Krait 400
 Memori : 3GB
 Sistem Operasi : Marshmallow
 Ukuran Layar : 5.2

5.2 Skenario Pengujian

Metode pengujian mengacu pada metode White-Box dan Black-Box. Pengujian akan dibagi menjadi dua bagian, yaitu pengujian fungsionalitas dan pengujian aplikasi terhadap pengguna. Pengujian ini dilakukan untuk mengetahui tolak ukur keberhasilan sistem dalam memenuhi fungsionalitas aplikasi.

5.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan menyiapkan beberapa skenario pengujian sebagai tolak ukur keberhasilan pengujian dan mengacu pada kasus penggunaan yang sudah dijelaskan pada sub bab 3.1.3. Metode pengujian yang digunakan

pada pengujian fungsionalitas mengacu pada metode White-Box dan Black-Box.

5.2.1.1 Pengujian Fitur *Log in*

Pengujian fitur *log in* merupakan pengujian terhadap aplikasi untuk menangani pengenalan pengguna untuk masuk ke dalam sistem menggunakan akun Gmail. Skenario pertama yaitu pengguna diminta untuk membuka aplikasi dalam keadaan tidak ada akun yang sedang aktif dalam aplikasi. Setelah itu, pengguna diminta menekan tombol “Sign In” dan memilih akun yang Gmail yang ingin digunakan. Pengujian ini juga dilakukan dengan dua kondisi yaitu saat *smartphone* terhubung dengan internet dan tidak terhubung dengan internet.

Hasil pengujian dari skenario pertama yaitu *smartphone* terhubung dengan internet dapat dilihat pada Tabel 5.1.

Tabel 5.1 Skenario Pengujian Fitur *Log in* ke-1

Nomor	UJ-0001-01
Referensi Kasus Penggunaan	UC-0001
Nama	<i>Log in</i>
Tujuan	Menguji sistem dalam menangani proses log in pengguna menggunakan akun Gmail.
Kondisi Awal	Pengguna membuka aplikasi.
Skenario	Pengguna menekan tombol “Sign In” dan memilih akun yang digunakan untuk masuk ke sistem.
Keluaran yang Diharapkan	Aplikasi menampilkan akun yang terintegrasi dan juga menyediakan tampilan untuk masuk dengan akun yang Gmail yang belum terdaftar pada <i>smartphone</i> . Aplikasi menampilkan halaman utama setelah pengguna memilih akun.

Tabel 5.1 Skenario Pengujian Fitur *Log in* ke-1

Hasil Pengujian	Berhasil dengan catatan
Kendala	Jika koneksi internet tidak stabil, maka aplikasi membutuhkan waktu lebih dari 2 detik untuk masuk ke menu utama.

Skenario kedua dilakukan tanpa ada koneksi internet pada *smartphone*. Hasil pengujian dari skenario ini dapat dilihat pada Tabel 5.2.

Tabel 5.2 Skenario Pengujian Fitur *Log in* ke-2

Nomor	UJ-0001-02
Referensi Kasus Penggunaan	UC-0001
Nama	<i>Log in</i>
Tujuan	Menguji sistem dalam menangani proses <i>log in</i> dengan keadaan <i>smartphone</i> yang tidak terhubung dengan internet.
Kondisi Awal	Pengguna membuka aplikasi.
Skenario	Pengguna menekan tombol “Sign In” dan memilih akun yang digunakan untuk masuk ke sistem.
Keluaran yang Diharapkan	Aplikasi menampilkan pesan kepada pengguna untuk memastikan koneksi terhubung dengan baik.
Hasil Pengujian	Berhasil
Kendala	-

5.2.1.2 Pengujian Fitur Mengirim Pesan

Pengujian fitur mengirim pesan merupakan pengujian terhadap aplikasi untuk mengirim pesan berupa gambar hasil konversi dari kanvas atau pesan berupa teks. Skenario pengujian dibagi menjadi empat tahapan. Dua skenario awal yaitu mengirim pesan melalui media *input* kanvas dalam keadaan *smartphone* terhubung dengan internet dan tidak, sedangkan dua

skenario lainnya yaitu mengirim pesan berupa teks dalam keadaan *smartphone* terhubung dengan internet dan tidak.

Hasil pengujian dari skenario pertama dapat dilihat pada Tabel 5.3.

Tabel 5.3 Skenario Mengirim Pesan Gambar ke-1

Nomor	UJ-0002-01
Referensi Kasus Penggunaan	UC-0002
Nama	Mengirim Pesan Gambar.
Tujuan	Menguji sistem dalam mengirim pesan yang diberikan pengguna melalui media input kanvas.
Kondisi Awal	Pengguna memilih lawan <i>chat</i> dan <i>smartphone</i> terhubung dengan internet.
Skenario	Pengguna menekan tombol “Send Canvas” dan memberikan pesan gambar, kemudian menekan tombol kirim.
Keluaran yang Diharapkan	Aplikasi menampilkan pesan terkirim pada daftar percakapan kedua pengguna.
Hasil Pengujian	Berhasil dengan catatan.
Kendala	Sistem memerlukan waktu untuk proses unggah gambar ke Firebase Storage.

Hasil pengujian dari skenario kedua dapat dilihat pada Tabel 5.4.

Tabel 5.4 Skenario Mengirim Pesan Gambar ke-2

Nomor	UJ-0002-02
Referensi Kasus Penggunaan	UC-0002
Nama	Mengirim Pesan Gambar.
Tujuan	Menguji sistem dalam mengirim pesan yang diberikan pengguna melalui media input kanvas.

Tabel 5.4 Skenario Mengirim Pesan Gambar ke-2

Kondisi Awal	Pengguna memilih lawan <i>chat</i> dan <i>smartphone</i> tidak terhubung dengan internet.
Skenario	Pengguna menekan tombol “Send Canvas” dan memberikan pesan gambar, kemudian menekan tombol kirim.
Keluaran yang Diharapkan	Aplikasi menampilkan pemberitahuan kepada pengguna untuk melakukan pengecekan koneksi.
Hasil Pengujian	Berhasil dengan catatan
Kendala	Sistem memerlukan waktu lebih lama untuk mengunggah gambar ke Firebase Storage ketika koneksi internet tidak stabil.

Hasil pengujian dari skenario ketiga dapat dilihat pada Tabel 5.5.

Tabel 5.5 Skenario Mengirim Pesan Gambar ke-3

Nomor	UJ-0002-03
Referensi Kasus Penggunaan	UC-0002
Nama	Mengirim Pesan Teks.
Tujuan	Menguji sistem dalam mengirim pesan teks yang diberikan pengguna.
Kondisi Awal	Pengguna memilih lawan <i>chat</i> dan <i>smartphone</i> terhubung dengan internet.
Skenario	Pengguna menekan tombol memasukkan pesan dan menekan tombol kirim.
Keluaran yang Diharapkan	Aplikasi menampilkan pesan terkirim pada daftar percakapan kedua pengguna.
Hasil Pengujian	Berhasil.
Kendala	-

Hasil pengujian dari skenario keempat dapat dilihat pada Tabel 5.6.

Tabel 5.6 Skenario Mengirim Pesan Gambar ke-4

Nomor	UJ-0002-04
Referensi Kasus Penggunaan	UC-0002
Nama	Mengirim Pesan Teks
Tujuan	Menguji sistem dalam mengirim pesan teks yang diberikan pengguna.
Kondisi Awal	Pengguna memilih lawan <i>chat</i> dan <i>smartphone</i> tidak terhubung dengan internet.
Skenario	Pengguna menekan tombol memasukkan pesan dan menekan tombol kirim.
Keluaran yang Diharapkan	Aplikasi menampilkan pemberitahuan kepada pengguna untuk melakukan pengecekan koneksi.
Hasil Pengujian	Berhasil.
Kendala	-

5.2.1.3 Pengujian Fitur Berbagi Kanvas

Pengujian fitur berbagi kanvas merupakan pengujian terhadap aplikasi untuk memberikan layanan berbagi kanvas kepada dua orang pengguna sehingga bisa bekerja dalam satu kanvas yang terhubung. Skenario pengujian dibagi menjadi dua. Skenario pertama yaitu kedua pengguna berbagi kanvas dan memberi sentuhan di kanvas pada *smartphone* yang terhubung dengan internet.

Hasil pengujian dari skenario pertama dapat dilihat pada Tabel 5.7.

Tabel 5.7 Skenario Pengujian Berbagi Kanvas ke-1

Nomor	UJ-0003-01
Referensi Kasus Penggunaan	UC-0003
Nama	Berbagi Kanvas
Tujuan	Menguji sistem menghubungkan dua pengguna melalui suatu kanvas (<i>Shareable Canvas</i>).
Kondisi Awal	Pengguna berada pada halaman <i>chat</i> dengan salah satu pengguna lain dan <i>smartphone</i> di kedua pengguna terhubung dengan internet.
Skenario	Kedua pengguna saling memberikan <i>gesture</i> pada masing-masing kanvas.
Keluaran yang Diharapkan	Aplikasi menyinkronisasikan kedua kanvas pengguna sehingga tampak sama di masing-masing <i>smartphone</i> .
Hasil Pengujian	Berhasil dengan catatan
Kendala	Apabila koneksi internet tidak stabil, maka kanvas di kedua pengguna memerlukan waktu lebih untuk sinkronisasi.

Skenario kedua yaitu salah satu pengguna memberikan sentuhan pada kanvas dalam keadaan *smartphone* tidak terhubung dengan internet. Setelah pengguna tersebut kembali terhubung dengan internet, maka aplikasi akan mengirim data ke Firebase Database sehingga kedua kanvas pengguna yang saling terhubung bisa tersinkronisasi.

Hasil pengujian dari skenario kedua dapat dilihat pada Tabel 5.8.

Tabel 5.8 Skenario Pengujian Berbagi Kanvas ke-2

Nomor	UJ-0003-02
Referensi Kasus Penggunaan	UC-0003
Nama	Berbagi Kanvas.
Tujuan	Menguji sistem menghubungkan dua pengguna melalui suatu kanvas (<i>Shareable Canvas</i>).
Kondisi Awal	Pengguna berada pada halaman <i>chat</i> dengan salah satu pengguna lain dan <i>smartphone</i> di salah satu pengguna tidak terhubung dengan internet.
Skenario	Pengguna A yang tidak terhubung dengan internet memberikan sentuhan, kemudian mengaktifkan koneksi <i>smartphone</i> -nya.
Keluaran yang Diharapkan	Kanvas pada pengguna B diperbarui setelah pengguna A menghubungkan kembali koneksi <i>smartphone</i> -nya.
Hasil Pengujian	Berhasil
Kendala	-

5.2.1.4 Pengujian Fitur Mencari Akun Pengguna Lain

Pengujian fitur mencari akun pengguna lain merupakan pengujian terhadap aplikasi untuk menampilkan daftar pengguna sesuai dengan masukan pengguna berupa nama atau *email*. Skenario yang dilakukan dalam pengujian ini dibagi menjadi tiga.

Skenario pertama yaitu pengguna memasukkan nama pengguna lain yang dicari dalam keadaan *smartphone* pengguna terhubung dengan internet. Hasil pengujian dari skenario pertama dapat dilihat pada Tabel 5.9.

Tabel 5.9 Skenario Pengujian Mencari Akun Pengguna ke-1

Nomor	UJ-0004-01
Referensi Kasus Penggunaan	UC-0004
Nama	Mencari Akun Pengguna Lain.
Tujuan	Menguji sistem untuk menampilkan daftar pengguna yang dicari oleh pengguna melalui <i>email</i> .
Kondisi Awal	Pengguna masuk ke halaman utama.
Skenario	Pengguna memilih tombol cari, kemudian mengisi nama pengguna yang ingin dicari.
Keluaran yang Diharapkan	Aplikasi menampilkan daftar pengguna yang namanya memiliki unsur dari kata yang dimasukkan sebagai kata kunci.
Hasil Pengujian	Berhasil
Kendala	-

Skenario kedua yaitu pengguna memasukkan *email* pengguna lain yang dicari dalam keadaan *smartphone* pengguna terhubung dengan internet. Hasil pengujian dari skenario kedua dapat dilihat pada Tabel 5.10.

Tabel 5.10 Skenario Pengujian Mencari Akun Pengguna ke-2

Nomor	UJ-0004-02
Referensi Kasus Penggunaan	UC-0004
Nama	Mencari Akun Pengguna Lain
Tujuan	Menguji sistem menampilkan daftar pengguna yang dicari oleh pengguna melalui <i>email</i> .
Kondisi Awal	Pengguna masuk ke halaman utama.
Skenario	Pengguna memilih tombol cari, kemudian mengisi <i>email</i> pengguna yang ingin dicari.

Tabel 5.10 Skenario Pengujian Mencari Akun Pengguna ke-2

Keluaran yang Diharapkan	Aplikasi menampilkan daftar pengguna yang <i>email</i> -nya memiliki unsur dari kata yang dimasukkan sebagai kata kunci.
Hasil Pengujian	Berhasil
Kendala	-

Skenario ketiga yaitu pengguna memasukkan *email* pengguna lain yang dicari dalam keadaan *smartphone* pengguna tidak terhubung dengan internet. Hasil pengujian dari skenario kedua dapat dilihat pada Tabel 5.11.

Tabel 5.11 Skenario Pengujian Mencari Akun Pengguna ke-3

Nomor	UJ-0004-03
Referensi Kasus Penggunaan	UC-0003
Nama	Mencari Akun Pengguna Lain
Tujuan	Menguji sistem menampilkan daftar pengguna yang dicari oleh pengguna melalui <i>email</i> .
Kondisi Awal	Pengguna masuk ke halaman utama dan perangkat tidak terhubung dengan internet
Skenario	Pengguna memilih tombol cari, kemudian mengisi <i>email</i> pengguna yang ingin dicari.
Keluaran yang Diharapkan	Aplikasi menampilkan pemberitahuan bahwa perangkat tidak terhubung dengan internet.
Hasil Pengujian	Berhasil
Kendala	-

5.2.1.5 Pengujian Fitur Menambahkan Teman

Pengujian fitur menambahkan teman merupakan pengujian terhadap aplikasi dalam melayani permintaan pertemanan antar pengguna. Skenario yang dilakukan dalam pengujian ini dibagi menjadi dua yaitu mengirim permintaan

pertemanan dalam keadaan *smartphone* pengguna yang terhubung dan tidak dengan internet.

Hasil pengujian dari skenario pertama dapat dilihat pada Tabel 5.12.

Tabel 5.12 Skenario Menambahkan Teman ke-1

Nomor	SP-UC-0005-01
Referensi Kasus Penggunaan	UC-0005
Nama	Menambahkan Teman.
Tujuan	Menguji sistem untuk mengirim permintaan pertemanan antar pengguna.
Kondisi Awal	Pengguna berada pada halaman cari pengguna dan telah memasukkan kata kunci berupa nama atau email pengguna yang ingin ditambahkan sebagai teman.
Skenario	Pengguna memilih akun pengguna yang ini ditambahkan, kemudian menekan tombol “Add”.
Keluaran yang Diharapkan	Akun yang ditambahkan terdaftar sebagai teman oleh pengguna yang menambahkan. Dan akun pengguna yang ditambahkan akan menerima permintaan pertemanan pada halaman utama.
Hasil Pengujian	Berhasil
Kendala	-

Hasil pengujian dari skenario kedua dapat dilihat pada Tabel 5.13.

Tabel 5.13 Skenario Menambahkan Teman ke-2

Nomor	UJ-0005-01
Referensi Kasus Penggunaan	UC-0005
Nama	Menambahkan Teman

Tabel 5.13 Skenario Menambahkan Teman ke-2

Tujuan	Menguji sistem untuk mengirim permintaan pertemanan antar pengguna.
Kondisi Awal	Pengguna berada pada halaman cari pengguna dan telah memasukkan kata kunci berupa nama atau <i>email</i> pengguna yang ingin ditambahkan sebagai teman. Keadaan <i>smartphone</i> tidak terhubung dengan internet.
Skenario	Pengguna memilih akun pengguna yang ini ditambahkan, kemudian menekan tombol “Add”.
Keluaran yang Diharapkan	Aplikasi menampilkan pemberitahuan bahwa perangkat tidak terhubung dengan internet.
Hasil Pengujian	Berhasil
Kendala	

5.2.1.6 Pengujian Fitur Mengonfirmasi Permintaan Pertemanan

Pengujian fitur mengonfirmasi permintaan pertemanan merupakan pengujian terhadap aplikasi dalam menangani konfirmasi pengguna terhadap permintaan pertemanan yang diterima. Skenario yang dilakukan dalam pengujian ini dibagi menjadi dua tahapan yaitu menerima permintaan pertemanan saat *smartphone* terhubung dan tidak dengan internet.

Hasil pengujian dari skenario pertama dapat dilihat pada Tabel 5.14.

Tabel 5.14 Skenario Mengonfirmasi Permintaan Pertemanan ke-1

Nomor	UJ-0006-01
Referensi Kasus Pengguna	UC-0006

Tabel 5.14 Skenario Mengonfirmasi Permintaan Pertemanan ke-1

Nama	Mengonfirmasi permintaan pertemanan.
Tujuan	Menguji sistem untuk melakukan konfirmasi pertemanan.
Kondisi Awal	Pengguna berada pada halaman utama dan <i>smartphone</i> terhubung dengan internet.
Skenario	Pengguna memilih permintaan pertemanan dan menekan tombol “Add”.
Keluaran yang Diharapkan	Aplikasi mengonfirmasi pertemanan dan menjadi akun yang dikonfirmasi sebagai teman pengguna aktif.
Hasil Pengujian	Berhasil
Kendala	

Hasil pengujian dari skenario kedua dapat dilihat pada Tabel 5.15.

Tabel 5.15 Skenario Mengonfirmasi Permintaan Pertemanan ke-2

Nomor	SP-UC-0006-02
Referensi Kasus Penggunaan	UC-0006
Nama	Mengonfirmasi permintaan pertemanan.
Tujuan	Menguji sistem untuk melakukan konfirmasi pertemanan.
Kondisi Awal	Pengguna berada pada halaman utama dan <i>smartphone</i> tidak terhubung dengan internet.
Skenario	Pengguna memilih permintaan pertemanan dan menekan tombol “Add”.
Keluaran yang Diharapkan	Aplikasi menampilkan pemberitahuan bahwa perangkat tidak terhubung dengan internet.
Hasil Pengujian	Berhasil
Kendala	

5.2.2 Pengujian Aplikasi terhadap Pengguna

Pengujian perlu dilakukan oleh pengguna secara langsung sehingga bisa mengetahui bahwa aplikasi telah memenuhi tujuan aplikasi atau belum, yaitu memberi pengalaman baru dalam berinteraksi dengan pengguna lain. Pengujian juga dilakukan untuk mengetahui bahwa fitur utama aplikasi telah berjalan sesuai dengan rancangan. Fitur utama yang diuji oleh pengguna pada bagian ini adalah pengiriman gambar sebagai hasil konversi dari kanvas yang pengguna buat dan membagi kanvas dengan pengguna lain sehingga bisa bekerja dalam satu kanvas yang terhubung.

Pengujian dilakukan oleh sepuluh orang yang diminta oleh penulis dan bersedia untuk melakukan pengujian. Pengguna diminta mengoperasikan aplikasi dengan beberapa skenario yang telah direncanakan oleh penulis. Skenario pengujian terhadap pengguna dibagi menjadi tiga tahapan yaitu sebagai berikut:

1. Pengguna A dan pengguna B secara bergantian saling mengirim gambar. Setiap gambar yang dibuat oleh salah satu pengguna harus ditebak oleh lawan *chat*. Masing-masing pengguna diberikan kesempatan memberi tebak gambar sebanyak tiga kali. Setelah melakukan tiga skenario di atas, pengguna juga diminta mengisi kuesioner yang dapat dilihat pada Tabel 5.16.
2. Dua orang pengguna diminta untuk memainkan permainan “Tic Tac Toe”. “Tic Tac Toe” adalah sebuah permainan yang dimainkan oleh dua orang. Aturan permainan adalah disediakan sebuah kotak sebesar 3 x 3 sebagai arena permainan. Masing-masing pengguna hanya bisa memberikan satu jenis tanda yaitu X atau O. Kesepakatan dilakukan di awal permainan, jika salah satu pengguna menggunakan tanda X, maka pengguna satu lagi hanya bisa menggunakan tanda O. Hal ini berlaku hingga setiap area

permainan terisi. Pengguna juga diminta mengisi kuesioner yang dapat dilihat pada Tabel 5.17.

3. Dua orang pengguna diminta untuk memainkan permainan “Dots and Boxes”. Disediakan sebuah arena permainan berupa titik-titik sejumlah 5×5 . Setiap pengguna secara bergantian membuat sebuah garis dengan menghubungkan dua titik yang bersebelahan. Ketika garis yang dibuat pengguna terhubung dengan garis lainnya dan membentuk sebuah persegi berukuran 1×1 . Maka pengguna tersebut mendapat skor 1. Pengguna juga diminta mengisi kuesioner yang dapat dilihat pada Tabel 5.17.

Masing-masing kolom pada kuesioner memiliki bobot penilaian yang berbeda dengan ketentuan sebagai berikut:

- Sangat setuju = 4
- Setuju = 3
- Tidak Setuju = 2
- Sangat Tidak Setuju = 1

Tabel 5.16 Hasil Pengujian Aplikasi terhadap Pengguna pada Fitur Mengirim Pesan

No	Pertanyaan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1	Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?	10	0	0	0

**Tabel 5.16 Hasil Pengujian Aplikasi terhadap Pengguna pada
Fitur Mengirim Pesan**

No	Pertanyaan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
2	Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?	5	5	0	0
3	Apakah aplikasi memberikan pengalaman baru dalam melakukan <i>chat</i> ?	7	3	0	0

**Tabel 5.17 Hasil Pengujian Aplikasi terhadap Pengguna pada
Fitur Berbagi Kanvas**

No	Pertanyaan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1	Apakah aplikasi menghubungkan kanvas milik anda	9	1	0	0

Tabel 5.17 Hasil Pengujian Aplikasi terhadap Pengguna pada Fitur Berbagi Kanvas

No	Pertanyaan	Pilihan Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
	dengan pengguna lain?				
2	Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?	3	7	0	0
3	Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur “Share Canvas”?	0	10	0	0
4	Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?	6	3	1	0

5.3 Evaluasi Pengujian

Tahap evaluasi akan dibagi menjadi dua bagian, yaitu evaluasi pengujian fungsionalitas dan evaluasi pengujian aplikasi terhadap pengguna.

5.3.1 Evaluasi Pengujian Fungsionalitas

Berdasarkan pengujian fungsionalitas yang dilakukan sebelumnya, dapat dilihat hasil dari semua pengujian menunjukkan nilai keberhasilan. Oleh karena itu dapat disimpulkan bahwa aplikasi sudah memenuhi semua fungsionalitas yang diharapkan.

5.3.2 Evaluasi Pengujian terhadap Pengguna

Pengujian yang dilakukan terhadap pengguna menghasilkan kesimpulan dalam berupa rata-rata yang menunjukkan persetujuan pengguna terhadap kesesuaian, kemudahan, dan ketertarikan fitur yang disediakan aplikasi. Kesimpulan hasil pengujian didapat dari rata-rata bobot jawaban atas pertanyaan kuesioner yang disediakan. Daftar jawaban pengguna dapat dilihat pada Tabel 5.18 dan Tabel 5.19.

Tabel 5.18 Evaluasi Pengujian terhadap Pengguna pada Fitur Mengirim Pesan.

No	Pertanyaan	Nilai
1	Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?	4,0

Tabel 5.18 Evaluasi Pengujian terhadap Pengguna pada Fitur Mengirim Pesan.

No	Pertanyaan	Nilai
2	Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?	3,5
3	Apakah aplikasi memberikan pengalaman baru dalam melakukan <i>chat</i> ?	3,7

Tabel 5.19 Evaluasi Pengujian terhadap Pengguna pada Fitur Berbagi Kanvas

No	Pertanyaan	
1	Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?	3,9
2	Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?	3,3
3	Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur “share canvas”?	3,0
4	Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?	3,5

Berdasarkan Tabel 5.18 dan Tabel 5.19 , dapat diketahui bahwa nilai rata-rata hasil kuesioner adalah 3,557. Nilai rata-rata yang telah melebihi angka 3 atau memiliki arti “setuju” menunjukkan bahwa aplikasi sudah nyaman dan mudah digunakan serta fitur pada aplikasi telah memberikan pengalaman baru kepada pengguna dalam berinteraksi dengan pengguna lain.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dijealaskan mengenai kesimpulan yang diambil dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga diberikan saran untuk pengembangan aplikasi kedepannya.

6.1 Kesimpulan

Dari hasil pengamatan selama proses pengerjaan yang dimulai dari tahapan perancangan, implementasi, dan pengujian yang telah dilakukan untuk membangun aplikasi, ditariklah kesimpulan-kesimpulan sebagai berikut:

1. Android Graphics Library mampu menyediakan modul-modul yang dibutuhkan dalam membuat sebuah kanvas pada pengembangan aplikasi Android seperti kelas Canvas, Paint, dan Path.
2. Firebase Realtime Database merupakan layanan yang digunakan pada aplikasi untuk menyimpan pesan antar pengguna. Layanan ini juga menyediakan layanan yang memungkinkan pengguna membaca data secara *realtime* sehingga sangat mendukung fitur *chat* pada aplikasi.
3. Firebase Storage merupakan layanan yang digunakan pada aplikasi untuk menyimpan gambar hasil konversi dari kanvas yang dikirim oleh pengguna. Firebase memungkinkan proses unduh dan unggah terhadap berkas yang pada tugas akhir ini berupa gambar.
4. Firebase Database mampu menghubungkan dua pengguna pada *node* khusus sehingga kedua pengguna melakukan pertukaran data secara *real time*. Hal ini membantu pengembangan aplikasi dalam memenuhi fitur berbagi kanvas yang secara otomatis melakukan pembaruan kanvas setiap ada aktivitas dari kedua pengguna.

6.2 Saran

Setelah melakukan pengembangan aplikasi, diharapkan aplikasi bisa dikembangkan jauh lebih baik nantinya. Beberapa saran untuk pengembangan aplikasi yang lebih baik ke depannya antara lain:

1. Melengkapi kanvas dengan variasi bentuk gambar yang lebih menarik seperti segitiga, trapesium, atau poligon.
2. Membuat kanvas menjadi lebih menarik dengan tambahan fitur saat menggambar seperti pilihan bentuk kuas yang lebih banyak.
3. Membuat kanvas yang bisa dibagikan ke lebih dari satu pengguna.

DAFTAR PUSTAKA

- [1] “Android (sistem operasi),” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 14-Des-2016.
- [2] “Mengenal Android Studio | Android Studio.”. Tersedia pada:
<https://developer.android.com/studio/intro/index.html?hl=id>. [Diakses: 15-Des-2016].
- [3] B. Octavianus, “[Android] Apa itu Firebase ?,” *coolnetkid*, 08-Sep-2016. .
- [4] “Java (programming language),” *Simple English Wikipedia, the free encyclopedia*. 25-Mei-2017.
- [5] “Canvas and Drawables | Android Developers.”. Tersedia pada:
<https://developer.android.com/guide/topics/graphics/2d-graphics.html>. [Diakses: 14-Des-2016].
- [6] “Realm Mobile Platform Overview.” [Daring]. Tersedia pada: <https://realm.io/docs/get-started/overview/>. [Diakses: 13-Jun-2017].

[Halaman ini sengaja dikosongkan]

LAMPIRAN A. HASIL KUISIONER

Nama : Arij N M
Pekerjaan : Mahasiswa
Kuisisioner

Kuisisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan *chat*?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju

Gambar A.7.1 Kuesioner Fitur Mengirim Pesan Pengguna 1

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
☐ Sangat setuju
☒ Setuju
☐ Kurang setuju
☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
☐ Sangat setuju
☒ Setuju
☐ Kurang setuju
☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
5. Saran untuk aplikasi

Untuk shareable, hasil share nya ditambah fitur send/save

Gambar A.7.2 Kuesioner Fitur Berbagi Kanvas Pengguna 1

Nama : IRZAL
 Pekerjaan : Mahasiswa
 Kuisioner

Kuisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.3 Kuesioner Fitur Mengirim Pesan Pengguna 2

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

Diberikan hapus sehingga dapat dibarat permanen layarnya
satur.

Gambar A.7.4 Kuesioner Fitur Berbagi Kanvas Pengguna 2

Nama : Wildan Lutfi SFS
Pekerjaan : Mahasiswa
Kuisisioner

Kuisisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju

Gambar A.7.5 Kuesioner Fitur Mengirim Pesan Pengguna 3

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
☐ Sangat setuju
☒ Setuju
☐ Kurang setuju
☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
☐ Sangat setuju
☒ Setuju
☐ Kurang setuju
☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
5. Saran untuk aplikasi

Gambar A.7.6 Kuesioner Fitur Berbagi Kanvas Pengguna 3

Nama : Aviananda
Pekerjaan : Mahasiswa
Kuisisioner

Kuisisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.7 Kuesioner Fitur Mengirim Pesan Pengguna 4

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☐ Sangat setuju
 - ☐ Setuju
 - ☒ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

Tambah tombol delete yang langsung menghapus kanvas. Atau diberi fitur penghapus untuk menghapus kanvas secara manual

Gambar A.7.8 Kuesioner Fitur Berbagi Kanvas Pengguna 4

Nama : *Nuzul Ayu Saffri*
Pekerjaan : *Mahasiswa*
Kuisisioner

Kuisisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.9 Kuesioner Fitur Mengirim Pesan Pengguna 5

Kuisiner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

Tambahan hapus gambar dan pengukuran pada bentuk π (preview gambar)

Gambar A.7.10 Kuesioner Fitur Berbagi Kanvas Pengguna 5

Nama : Damai Matsa B
 Pekerjaan : Mahasiswa TC
 Kuisioner

Kuisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.11 Kuesioner Fitur Mengirim Pesan Pengguna 6

Kuisisioner Fitur "Shareable Canvas"

- Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
- Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
- Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
- Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
- Saran untuk aplikasi

- waktu selesai mengetik keyboard gambar ditutup harus diback dulu.

- dikasih fitur erase

- backgroundnya jangan cuma putih aja, kalau bisa diwarnain juga.

Gambar A.7.12 Kuesioner Fitur Berbagi Kanvas Pengguna 6

Nama : Rizky Fernaldo M
Pekerjaan : Mahasiswa
Kuisisioner

Kuisisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
☒ Sangat setuju
☐ Setuju
☐ Kurang setuju
☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
☐ Sangat setuju
☒ Setuju
☐ Kurang setuju
☐ Tidak setuju

Gambar A.7.13 Kuesioner Fitur Mengirim Pesan Pengguna 7

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

Ditambah fitur hapus dan notif ketika user lain bergabung.

Gambar A.7.14 Kuesioner Fitur Berbagi Kanvas Pengguna 7

Nama : Ahmad Ismail H. W
 Pekerjaan : Mahasiswa
 Kuisioner

Kuisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.15 Kuesioner Fitur Mengirim Pesan Pengguna 8

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

- kasih tombol hapus untuk menghapus goresan pada kanvas
 shg tidak undo /redo berkali-kali

Gambar A.7.16 Kuesioner Fitur Berbagi Kanvas Pengguna 8

Nama : Nabilah Zaki Lismia
 Pekerjaan : Mahasiswa
 Kuisioner

Kuisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.17 Kuesioner Fitur Mengirim Pesan Pengguna 9

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

Lebih diperbanyak fiturunya, misal ~~sa~~ multi-chat gitu.

Gambar A.7.18 Kuesioner Fitur Berbagi Kanvas Pengguna 9

Nama : Anfar
Pekerjaan : Mahasiswa
Kuisisioner

Kuisisioner Fitur "Chat"

Beri tanda X pada kotak pilihan yang anda pilih

1. Apakah pesan gambar/teks yang dikirim oleh pengguna lain kepada anda dapat diterima dengan baik?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah tampilan aplikasi mempermudah dan membuat anda nyaman dalam menggunakan aplikasi?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah aplikasi memberikan pengalaman baru dalam melakukan chat?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju

Gambar A.7.19 Kuesioner Fitur Mengirim Pesan Pengguna 10

Kuisisioner Fitur "Shareable Canvas"

1. Apakah aplikasi menghubungkan kanvas milik anda dengan pengguna lain?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
2. Apakah kanvas secara cepat diperbarui ketika pengguna lain yang terhubung membuat suatu gambar/goresan?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
3. Apakah tampilan aplikasi mempermudah anda dalam menggunakan fitur "share canvas"?
 - ☐ Sangat setuju
 - ☒ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
4. Apakah aplikasi memberikan pengalaman baru kepada anda dalam berinteraksi kepada pengguna?
 - ☒ Sangat setuju
 - ☐ Setuju
 - ☐ Kurang setuju
 - ☐ Tidak setuju
5. Saran untuk aplikasi

- ketebalan garis dapat diubah

- tambahkan fitur penghapus

Gambar A.7.20 Kuesioner Fitur Berbagi Kanvas Pengguna 10

BIODATA PENULIS



Luffi Aditya Sandy, lahir di kota Bengkulu pada tanggal 30 November 1995. Penulis menempuh pendidikan mulai dari SD Negeri 08 kota Bengkulu (2001-2007), SMP Negeri kota Bengkulu (2007-2010), SMA Negeri 5 kota Bengkulu (2010-2013) dan masuk perkuliahan di jurusan Teknik Informatika pada tahun 2013 dan saat ini masih menempuhnya. Penulis memiliki ketertarikan pada kegiatan olahraga seperti Sepak Bola. Selain itu penulis juga tertarik dalam hal-hal baru di dunia IT. Di jurusan Teknik Informatika, penulis menjadi salah satu administrator di Laboratorium Algoritma dan Pemrograman dan memilih rumpun mata kuliah Algoritma dan Pemrograman. Selain itu, penulis juga memiliki ketertarikan pada pemrograman perangkat bergerak dan teknologi pendukungnya. Selama perkuliahan, penulis aktif dalam beberapa kegiatan dan organisasi kemahasiswaan, antara lain, staff Departemen Pengembangan Profesi Himpunan Teknik Computer-Informatika tahun 2014-2015, staff Divisi Bina Muslim Keluarga Muslim Informatika 2014-2015, wakil ketua National Logic Competition Schematics 2014-2015, ketua Divisi Hubungan Kelembagaan Keluarga Muslim Informatika tahun 2015-2016, staff ahli Departemen Pengembangan Profesi Himpunan Teknik Computer-Informatika tahun 2015-2016, dan staff teknis National Logic Competition 2015-2016. Penulis juga berkesempatan menjadi asisten dosen di beberapa matakuliah jurusan Teknik Informatika yaitu Perancangan Analisis Algoritma, Pemrograman Berorientasi Objek, dan Dasar Pemrograman. Penulis dapat dihubungi melalui alamat *email*: luffi.as@gmail.com.